Inférence des acteurs de la régulation des expressions géniques

Laetitia Bourgeade

université de BORDEAUX

Thèse présentée
pour obtenir le grade de

Docteur de
l'Université de Bordeaux

École Doctorale de Mathématiques et
d'Informatique
Spécialité Informatique

par Laetitia BOURGEADE

Inférence des Acteurs de la Régulation des Expressions Géniques.

Sous la direction de : Julien ALLALI
(Co-directrice : Élisabeth BON)

Soutenue le 30/01/2015

Membres du jury :

Julien ALLALI	Maître de Conférences (INP-LaBRI)	Directeur
Guillaume BLIN	Professeur des Universités (Université de Bordeaux-LaBRI)	Président
Élisabeth BON	Maître de Conférences (Université de Bordeaux-LaBRI)	Co-directrice
Alain DENISE	Professeur des Universités (Université Paris Sud-LRI)	Examinateur
Christine GASPIN	Directrice de Recherches (INRA Toulouse)	Rapporteur
Hélène TOUZET	Directrice de Recherches (CRISTAL-INRIA Lille)	Rapporteur

Titre Inférence des Acteurs de la Régulation des Expressions Géniques.

Résumé La quantité croissante de données générées est à l'origine de nombreuses problématiques en bioinformatique telles que le développement de nouvelles méthodes de traitement et d'analyse efficaces de ces données. Plus particulièrement, les réseaux de régulation des fonctions cellulaires sont au coeur de nombreux projets aujourd'hui. Il est donc nécessaire, afin d'appréhender correctement ces systèmes de régulation, de comprendre l'origine et de caractériser les acteurs de ces systèmes tels que les ARN et les pseudogènes.
Nous avons établi une nouvelle méthode de comparaison d'une séquence ARN requête avec un jeu de séquences ARN cibles. Notre méthode se base sur (i) l'indexation préalable des graines en séquence/structure des ARN du jeu cible, (ii) la recherche des ARN cibles par détection des graines de la séquence requête présentes également dans le jeu de données cible et le chainage de ces graines, puis (iii) la complétion de l'alignement obtenu à l'aide d'un algorithme d'alignement exact incorporant des contraintes d'alignement. Cette méthode a été appliquée sur le jeu de données de BraliBase2.1. L'exactitude des résultats obtenus et l'efficacité de la méthode ont alors été comparés à la méthode d'alignement exact *LocARNA* et à son filtre basé sur un algorithme de chainage de graines récemment développé, *ExpLocP*. Notre méthode *RNA-unchained* permet d'améliorer significativement les temps de calcul de *LocARNA* et présente des temps de calcul similaires à *ExpLocP*, tout en améliorant l'exactitude des alignements finaux.
De plus, nous avons développé une méthode, *PseudOE*, de détection et de caractérisation du pseudome au sein d'un génome et d'analyse comparative de ce pseudome entre plusieurs génomes. Cette méthode a ainsi permis de réaliser l'analyse du panpseudome de deux souches relativement distantes de l'espèce *Oenococcus oeni* et qui présentent des propriétés oenologiques opposées. On observe dans ces génomes compacts, de 1,8Mb, 8,5% de pseudogènes. Par comparaison aux autres génomes bactériens, les génomes d'*O. oeni* semblent sensibles à la pseudogénisation. La majorité des pseudogènes détectés ont pour origine des mutations de leur séquence et sont présents uniquement dans l'un des génomes, ce qui soutient l'hypothèse d'une origine récente de ces séquences et qui illustre la tendance des *O. oeni* à l'hypermutabilité. De plus, l'analyse des données fournies par *PseudOE* a permis la mise en évidence d'une organisation spatiale des pseudogènes au sein de territoires spécifiques du chromosome. L'ensemble de ces analyses illustre les particularités des pseudogènes chez *O. oeni* et apporte des informations supplémentaires concernant l'évolution des gènes/génomes dont les annotations de génomes pourraient retirer des bénéfices.

Mots-clefs ARN, stucture secondaire, indexation, filtrage, alignement, graines, chaînage, comparaison (un *vs.* plusieurs), similarité, pseudogènes, pseudome, comparaisons génomiques, évolutions génomiques, plasticité, adaptation, *Oenococcus oeni*.

Title The Inference of Gene Expression Regulator actors.

Abstract The increasing amount of available data is a source of many issues in bioinformatics such that the development of new methods of treatments and efficient analysis of data. Especially, regulatory networks are at the heart of many projects. Also, in order to understand regulatory systems, it appears to be necessary to characterize and to understand actors of these systems such as RNA and pseudogenes. We develop a new method to compare a query RNA with a static set of target RNAs. Our method is based on (i) a preliminary indexing of the sequence/structure seeds of the target RNAs, (ii) searching the potentially homolog RNAs by detecting seeds of the query present in targets, chaining these seeds, then (iii) completing the alignment using an anchor-based exact alignment algorithm. We apply our method on the benchmark Bralibase2.1. We compare our method accuracy and efficiency with the exact method *LocARNA* and its recent seeds-based speed-up *ExpLocP*. Our pipeline *RNA-unchained* greatly improves computation time of *LocARNA* and is comparable to the one of *ExpLocP*, while improving the overall accuracy of the final alignments.
Moreover, we develop a new method, *PseudOE*, to detect and to characterize the pseudome of one genome, and to analyse by comparison two genomes at least. This method allows to analyse the pan-pseudome of two distantly related *Oenococcus oeni* strains with opposite oenological properties. Quite interestingly, with 8.5% of pseudogenes for a compact 1.8Mb genome, *O. oeni* appeared to be prone to pseudogenization compared to other bacteria. A great proportion of pseudogenes were found to come from mutational degradation suggesting a relatively recent origin that could illustrate the natural propensity of *O. oeni* for hypermutability. In addition, we identify a spatial organization of pseudogenes into dedicated chromosomal territories. These analysis illustrate peculiar properties of *O. oeni* pseudogenes, providing additional insights of gene/genome evolution from which future genome annotation will benefit.

Keywords RNA, secondary stucture, indexing, filtering, alignment, seeds, chaining, one *vs.* all comparisons, similarity, Pseudogenes, Pseudome, Comparative genomics, Genome evolution, Gene plasticity, Niche adaptation, Oenococcus oeni.

Laboratoire Bordelais de Recherche en Informatique - UMR 5800

Les machines un jour pourront résoudre tous les problèmes, mais jamais aucune d'entre elles ne pourra en poser un !

Albert Einstein

Remerciements

Il m'était très difficile de ne pas remercier tous ceux que j'ai côtoyés lors de ces années de thèse car sans leur concours, leur soutien et leur générosité j'aurais eu bien du mal à mener ces travaux à leur terme.

En premier lieu je tiens à remercier mes directeurs de thèse, Julien Allali et Elisabeth Bon, pour ces trois années qui ne furent finalement pas si horribles grâce à leur implication, leurs conseils, leur disponibilité et leur tolérance. Leur professionnalisme et leurs qualités m'ont permis d'élargir mes compétences et de m'intéresser à de nouveaux problèmes. Je remercie Elisabeth de m'avoir soutenue au cours de ces trois années. Je remercie Julien Allali de m'avoir permis de rencontrer des personnes comme Cédric Chauve et Yann Ponty mais surtout d'avoir su être présent aux moments nécessaires et quelle que soit l'heure.

Je remercie mes rapporteurs, Christine Gaspin et Hélène Touzet, pour avoir accepté d'effectuer cette tâche dans des délais très serrés et pour l'attention avec laquelle ils ont lu le présent manuscrit.

Je remercie également Alain Denise d'avoir accepté d'être membre de mon jury et pour le temps qu'il a consacré à mes travaux.

Je remercie Guillaume Blin pour son implication active dans la préparation de ma soutenance de thèse et des formalités qui l'accompagnent, tout autant que pour avoir consacré du temps à la lecture de mes travaux et avoir accepté d'être un membre de mon jury.

Mes remerciements vont également à Cédric Chauve qui, à deux reprises, m'a accueillie au sein de son équipe au département de mathématiques de SFU à Vancouver. Je le remercie particulièrement pour m'avoir fait redécouvrir la bioinformatique par ses approches particulières, pertinentes et efficaces.

Je remercie également Yann Ponty pour tous les échanges que nous avons eu lors de mon dernier séjour au Canada et à chaque fois que nous nous sommes recroisés par la suite, mais aussi pour m'avoir hébergée et ainsi permis d'être à l'heure pour mon vol retour, sans avoir à dormir dans l'aéroport.

Je remercie Georges Eyrolles de m'avoir fait découvrir des subtilités du langage java, pour ses mémorables réunions de préparation des TD, mais surtout de m'avoir accueillie chaleureusement ces derniers mois dans son bureau. Je le remercie également, avec Julien, de m'avoir distraite au cours de ma rédaction avec leur fresque géante à colorier. Une fresque qui, malheureusement, démarrait plus vite que mon manuscrit mais qui, au final, n'est pas encore terminée, contrairement à ce manuscrit.

Je remercie également Patricia Thébault, Pascal Desbarats, Bruno Pinaud, David Auber, Marie Beurton Aimar, Isabelle Dutour et Tiphaine Martin pour tout le soutien et l'aide apportée dans l'organisation et les avancées de mon travail.

Je remercie mes co-galériennes de thèse et co-bureau, qui sont maintenant devenues de vraies amies,Razanne Issa et Louise-Amélie Schmitt, pour leur soutien et les nombreux échanges que nous avons eu aussi bien sur nos sujets de thèses, aussi éloignés les uns des autres soient-ils, que sur des sujets plus légers.

Au cours de ces années j'ai fait partie de l'équipe MaBioVis, je remercie chacun des membres avec lesquels j'ai pu échanger sur de nombreux domaines et dont les remarques ont pu me permettre d'envisager certains problèmes sous un nouvel angle.

Je remercie pour leur gentillesse tout le personnel du LaBRI, Maité, Sylvie, Catherine, Philippe, Brigitte et Luce, et Gaëlle de l'INP-ENSEIRB-Matmeca, toujours présents quand on a besoin d'aide aussi bien pour relier un rapport que pour monter un dossier. Sans oublier les membres de l'équipe systèmes qui m'ont plus d'une fois aidée.

Heureusement que mes amis étaient là aussi pour me changer les idées. Merci Angélique, Laura, Marianne, Noémie, Julien, François, Boris, Florian, Lionel et Camille pour toutes ces sorties, ces soirées ou ces chocolats chauds partagés avec vous. La thèse aura au moins eu comme conséquence de resserrer certains lien d'amitié...

Mais je n'aurais sûrement pas tenu ces trois années sans Sand, Caro, Perrine, Jody, Margot, Myriam, Jerem, Lauryne, Juliette et Sasha qui chaque semaine me permettaient de m'échapper à travers la danse et partageaient tellement à travers chacune des représentations d'Akadanse.

Je remercie mes parents et ma soeur pour d'innombrables raisons, la moindre d'entre elles étant d'avoir à plusieurs reprises relu ce manuscrit.

Enfin je remercie Guillaume d'avoir supporté et accepté les soirées et nuits de travail, mes départs pour diverses conférences et le Canada mais qui a aussi su me faire traverser les inévitables périodes de découragement apparues au cours de cette thèse.

Finalement, je remercie et demande pardon à tous ceux que j'ai malheureusement oubliés...

Table des matières

Introduction **9**

1 Contexte **21**
1.1 La Cellule : Unité du Vivant 21
1.1.1 Généralités sur la Cellule 21
1.1.2 Les Processus d'Expression de l'Information Génétique 24
1.1.3 Le Gène Unité de Base des Fonctions Cellulaires 35
1.2 L'Évolution des Génomes 43
1.2.1 Mécanismes de Divergence des Génomes 43
1.2.2 Dynamiques d'Évolution des Génomes 46
1.3 Les Objets Biologiques Non Codants 49
1.3.1 Les Acides RiboNucléiques ou ARN 49
1.3.2 Les Pseudogènes 53

2 Comparaisons **65**
2.1 Comparaison de Séquences 65
2.1.1 Modélisation, Notations et Définitions 66
2.1.2 Le Problème d'Alignement Global Optimal 71
2.2 Chaînage dans les Séquences 76
2.2.1 Chaînage 1D en Séquence 76
2.2.2 Chaînage 2D en Séquences 77
2.2.3 Algorithme Hybride de Chaînage 2D en Séquences 82
2.3 Deux Principaux Filtres d'Alignement 91
2.3.1 Algorithme FastA 91
2.3.2 Algorithme BLAST 95

3 ARN : Filtrage & Indexation **101**
3.1 Repliement, Modélisation et Comparaison 101
3.1.1 Modélisation, Notations et Définitions 101
3.1.2 Problème de Comparaison d'ARN Deux à Deux 107
3.2 Chainage dans les Arborescences 113
3.2.1 Définitions, Préliminaires et Établissement du Problème . . . 113
3.2.2 Deux Algorithmes de Chainage 2D dans les Arborescences . . 115

3.3 Filtre pour la Comparaison d'Arborescences . . . 117
3.3.1 Modélisation et Indexation des Graines . . . 118
3.3.2 Recherche de Similitudes de Structures Secondaires d'ARN . . 123
3.4 Tests et Performance du Filtre . . . 129
3.4.1 Outils d'Analyse et de Comparaison . . . 130
3.4.2 Analyses Comparatives de l'Impact des Différentes Graines Sélectionnées . . . 133
3.4.3 Impact des Options des Graines . . . 140
3.5 Conclusion & Perspectives . . . 144

4 Pseudogènes : Identification & Caractérisation 151
4.1 Nomenclature Systématique des Objets Génétiques . . . 153
4.1.1 Nécessité d'une Nomenclature Commune . . . 153
4.1.2 Élaboration d'une Nomenclature Systématique . . . 153
4.2 Caractérisation des Pseudogènes . . . 156
4.2.1 Caractérisation . . . 156
4.2.2 Recensement . . . 158
4.3 Procédure d'Identification des Pseudogènes . . . 163
4.3.1 Consolidation des Données d'Origine . . . 164
4.3.2 Identification des Blocs de Séquences Candidates . . . 165
4.3.3 Identification des Pseudogènes Potentiels . . . 166
4.3.4 Relations Phylogénétiques . . . 167
4.3.5 Performances de *PseudOE* . . . 169
4.4 Analyse Comparative et Topologique . . . 170
4.4.1 Inventaire des Pseudogènes . . . 170
4.4.2 Populations Pseudogéniques : Plasticité Génique . . . 171
4.4.3 Plasticité du Pseudome et Évolution . . . 174

Synthèse & Perspectives 183

Annexes 187

Table des figures

1.1 Compaction de l'ADN . . . 22
1.2 Cellules eucaryote et procaryote . . . 23
1.3 Désoxyribonucléotides . . . 24
1.4 Brin d'ADN . . . 25
1.5 Diffraction rayon X . . . 26
1.6 Bases complémentaires de l'ADN . . . 26
1.7 Double hélice ADN . . . 27
1.8 Transcription . . . 28
1.9 Ribonucléotides . . . 30
1.10 Code génétique . . . 31
1.11 Phase de lecture . . . 32
1.12 Traduction . . . 33
1.13 Liaison peptidique . . . 34
1.14 Expression des gènes . . . 36
1.15 Gène eucaryote . . . 37
1.16 gène procaryote . . . 38
1.17 Familles d'ARN . . . 39
1.18 Régulation de l'expression génique procaryote . . . 42
1.19 Régulation de l'expression génique eucaryote . . . 42
1.20 Orthologues . . . 47
1.21 Brin d'ARN . . . 49
1.22 Paires de bases d'ARN . . . 50
1.23 Structures secondaires d'ARN . . . 51
1.24 Evolution de la taille de la Rfam . . . 53

2.25 Matrices nucléiques . . . 69
2.26 Propriétés physico-chimiques des acides aminés . . . 71
2.27 chaînage 1D en séquence . . . 76
2.28 chaînage 2D en séquence . . . 78
2.29 Arbre binaire . . . 85
2.30 Algorithme hybride . . . 88
2.31 Algorithme FastA . . . 93
2.32 Algorithme BLAST . . . 96

2.33 Architecture filtre 98

3.34 Structure arc-annotée 104
3.35 Graphe de corde 104
3.36 Représentations arborescentes 106
3.37 Opérations d'édition 108
3.38 Comparaison d'arborescences par édition 109
3.39 Comparaison d'arborescence par alignement 110
3.40 Chaînage dans arborescence 116
3.41 Architecture RNA-unchained 117
3.42 Graines (l,d) centrées 119
3.43 Pré hits 120
3.44 Architecture RNA-unchained 120
3.45 Architecture RNA-unchained 122
3.46 Pré Hits : option r 123
3.47 Architecture RNA-unchained 123
3.48 Complétion des pré hits 124
3.49 Compatibilité des hits : ancestralité 126
3.50 Hits Compatibles 127
3.51 Architecture RNA-unchained 127
3.52 Extension des hits 128
3.53 Architecture RNA-unchained 129
3.54 Exemple d'ancre 129
3.55 Courbes types 132
3.56 Shapes 134
3.57 Influence méthode de repliement : MFE 135
3.58 Influence méthode de repliement : MEA 136
3.59 Influence méthode de repliement : Shapes 136
3.60 Influence méthode de repliement : multi-structures 137
3.61 Influence de la taille de la séquence 137
3.62 Influence de la taille de la structure 138
3.63 Influence des options d'*RNA-unchained* 139
3.64 Comparaison *RNA-unchained*, *LocARNA* et *ExpLocP* 139
3.65 Contraintes communes 141
3.66 Nombre d'alignements couverts 142

4.67 Nomenclature 154
4.68 Types de pseudogènes 157
4.69 Modèle d'évolution des gènes 158
4.70 Phylogénie Bactéries Lactiques 160
4.71 Pipeline PseudOE 163
4.72 Étape de ré-annotation 164
4.73 Identification des Blocs 165

4.74 Étape d'identification des pseudogènes potentiels 166
4.75 Étape de caractérisation des relations phylogénétiques 168
4.76 Pattern de comparaison 169
4.77 Fonctions pseudogénisées 172
4.78 Topologie de répartition des pseudogènes 173
4.79 Distribution des pseudogènes 175
4.80 Cycle de pseudogénisation 177

81 Diverses tailles de graines 189
82 Impact des options d'RNA-unchained et taille constante en séquence . 189
83 Impact des options d'RNA-unchained et taille constante en structure 190
84 Proportions des pseudogènes 191

Liste des tableaux

1.1 Familles ARN 39
1.2 Prévalence Pseudogènes 54

3.3 Temps de calcul 143

4.4 Comparaison des données génomiques 171
4.5 Proportions des pseudogènes 172
4.6 Architecture du pseudome 176

7 Prévalence Pseudogènes Lactobacilliales 188
8 Table de Correspondances 192
9 Table de Correspondances(2) 193
10 Table de Correspondances(3) 194

Liste des algorithmes

1 Comparaison 2D en séquence : programmation dynamique 80
2 Comparaison 2D en séquence : balayage 81
3 Score de chaînage 85
4 Meilleur score de chaînage 86
5 Algorithme hybride 89

Introduction

L'analyse et la compréhension des processus biologiques qui régissent la vie, et en particulier du génome qui assure la pérennité et la transmission des propriétés des cellules et des organismes, requièrent le développement d'outils informatiques dédiés et performants. C'est dans ce contexte que les sciences de la vie se sont rapprochées des sciences formelles telles que les mathématiques et l'informatique. Ces interactions permettent de traiter les immenses masses de données générées par les avancées technologiques de ces dernières décennies. Mais pour que cette concertation entre disciplines mène à de meilleures analyses et modélisations, les efforts d'interdisciplinarité entrepris doivent être poursuivis.

Les techniques de séquençage ont évolué vers une nouvelle génération de machines dites à très haut débit. On parle de « Next Generation Sequencing ». Des centaines de Gigabases sont séquencées par semaines à des coûts bien moindre qu'au début du séquençage et à partir seulement de peu de matériel génétique initial. Cette révolution offre des opportunités pour le développement d'applications sur l'analyse et la caractérisation de telles séquences mais aussi sur la compréhension de tous les mécanismes sous-jacents. En effet, si au début de la génétique et des méthodes automatiques d'analyse des génomes on s'attachait principalement au séquençage des séquences géniques, de nos jours la compréhension des mécanismes régissant l'expression de ces gènes constitue l'un des principaux axes d'analyse. C'est dans ce contexte actuel d'analyse dynamique que la biologie de synthèse a émergé. En effet, afin de concevoir des organismes de synthèse qui pourraient apporter des thérapies plus efficaces, de nouveaux matériaux facilement recyclables, des biocarburants, ..., la connaissance des séquences géniques n'est pas suffisante pour (re)créer de telles voies cellulaires. La maîtrise des éléments de régulation constitue un point clef de l'élaboration de tels organismes.

Au delà du séquençage des génomes, les progrès techniques ont ouvert la voie à de nombreux champs d'études parmi lesquels on dénombre l'analyse et la quantification de l'expression génique (*RNAseq*), la détection des interactions Acide Désoxyribo-Nucléique (ADN)/Acide RiboNucléique (ARN)-protéines (*ChipSeq*), ...Ces progrès techniques se répercutent sur toutes les recherches qui en découlent et de nouvelles approches basées sur ces grandes quantités de données émergent via la bioinformatique.

Face à ce volume croissant de données complexes et hétérogènes, la bioinformatique comparative et prédictive doit relever un double défi : du point de vue du

volume de données à traiter, qui peut présenter un enjeu algorithmique, et du point de vue des données séquencées elles-mêmes, puisque pour tout nouveau type de données les outils existants ne sont pas nécessairement transposables directement. L'analyse des données expérimentales associée au développement de nouveaux algorithmes reste donc un enjeu actuel.

L'accès à cet important volume de données combiné à la découverte d'un nombre croissant d'éléments dits non codant et présentant un rôle fonctionnel a révolutionné le dogme central de la génétique centré jusqu'alors autour du rôle prépondérant des gènes codant pour des protéines. Cette face immergée du génome héberge, entre autres, les pseudogènes longtemps répertoriés comme « ADN poubelle ». Cette qualification a récemment été remise en cause par des découvertes les désignant comme des leurres fonctionnels de part leur activité de régulateur de l'expression génique. Les pseudogènes dérivent d'objets codants ce qui présente un intérêt informatique du point du vue de l'identification de ces entités via l'analyse de leur syntaxe. Ils constituent en effet des leurres linguistiques difficiles à détecter de manière automatique au sein du génome et donc à annoter.

Ces régions dites non-codantes hébergent aussi des gènes à ARN. Une des avancées majeures de ces dernières années en biologie moléculaire a été la découverte de nouvelles familles d'ARN non codants (ARNnc). Ces séquences ont longtemps été perçues comme de simples intermédiaires dans le décodage de l'information génétique, les dernières avancées montrent qu'elles sont impliquées dans de nombreux processus cellulaires, via leur conformation spatiale, tels que la régulation de l'expression génique, l'épissage, ... (Cech et al., 1992). Cela est particulièrement bien illustré par la croissance de la base de données Rfam (Burge and al., 2013), dont la taille est passée de 15 255 ARN en 2002 (date de sa création), à 19 623 515 en 2014 (dernière mise à jour). De plus, de nombreuses études récentes du structurome ARN, soit de l'ensemble des structures répertoriées, ont permis la découverte de nouvelles familles d'ARNnc et une meilleure compréhension du rôle des ARN dans la cellule (Will et al., 2007; Kertesz et al., 2010; Wan et al., 2011).

Ces deux entités, ARN et pseudogènes, actrices dans la régulation de l'expression génique constituent un espace de recherches à explorer et décrypter. Nos recherches se sont donc focalisées sur l'analyse de ces objets.

Un premier axe s'articule autour de la généralisation des méthodes de filtrage des données structurées, que représentent les ARN dans les grandes bases de données, permettant de détecter efficacement les ARN d'un génome par le développement d'outils de comparaison d'ARN et d'outils de comparaison de structures secondaires d'ARN. Pour cela, un filtre efficace de recherche d'ARN, inspiré des filtres pour séquences BLAST et FastA, est développé pour la fouille des grandes bases de données. Le défi réside dans la modélisation de graines sur des arborescences, l'indexation de ces graines et l'exploitation des occurrences comptabilisées.

Les filtres reposent sur une méthodologie commune d'identification de motifs, d'étapes de raffinement et de sélection de ces motifs et enfin d'une comparaison finale. Comparer les séquences entre elles permet alors d'obtenir des informations

sur le degré de similarité, ou au contraire de différence, entre ces séquences. Ces approches ont pour intérêt de pouvoir classifier les séquences, voire même de les identifier, mais elles permettent également de retracer l'histoire évolutive qui unit ces séquences et, par là même, celle qui relit les espèces qui les possèdent. La comparaison de séquences est donc le point de départ des analyses portant sur la classification ou sur l'établissement de phylogénies.

L'algorithme d'alignement de séquences, méthode privilégiée pour la comparaison, a une complexité quadratique (Gusfield, 1997). Cette complexité a amené à développer des méthodes de comparaison moins coûteuses comme le chaînage de motifs communs. Les comparaisons permettent de mettre en avant les régions les plus semblables et donc de déterminer si des régions sont communes ou bien incluses dans des régions plus grandes. Il apparaît donc comme important de s'intéresser à ces motifs. Afin de réaliser de telles analyses comparatives, l'utilisation des bases de données mises en place se révèle utile mais laborieuse. En effet, les méthodes de comparaison deux à deux présentent une complexité au minimum cubique et ne sont pas optimisées pour des analyses à grande échelle. C'est pourquoi il semble intéressant de pouvoir filtrer efficacement ces bases de données, sur la base de régions conservées, afin d'en extraire les données recherchées.

Le principe de cette technique repose (1) sur la détection et le calcul du score des motifs communs les plus conservés, on appellera ces motifs *hits*, (2) sur le calcul du sous-ensemble de score maximal de hits colinéaires non chevauchants dans chacune des deux séquences considérées (ou *chaîne optimale*). Cette chaîne optimale sert alors de squelette à l'alignement des deux séquences, et seuls les fragments entre les hits sélectionnés du squelette sont alignés. Ce type d'approche est utilisé par de nombreux programmes d'alignement de séquences tels que *FastA* (Lipman and Pearson, 1985) ou *MGA* (Höhl et al., 2002) pour n'en citer que deux.

Lors de l'analyse des filtres en séquences, nous nous sommes intéressés au problème du calcul d'une chaîne optimale de hits à partir d'un jeu de k hits sur deux séquences S_1 et S_2, de longueurs respectives n et m. Ce problème peut être résolu en $O(k + n \times m)$ en temps par un algorithme de programmation dynamique. Cependant, en pratique, le nombre de hits k est naturellement subquadratique, ce qui motive l'élaboration d'algorithmes dits par « balayage » dont la complexité dépend uniquement de k : $O(k \log k)$ en temps (Ohlebusch and Abouelhoda, 2006).

La combinaison de ces deux principaux algorithmes permettrait-elle de tirer avantage de chacun d'eux ? En effet, théoriquement, le nombre de hits k peut être quadratique et sous certaines conditions $O(k \log k)$ peut être plus élevé que $O(k+n\times m)$. Une approche naïve consisterait à comparer $k+n\times m$ et $k \log k$ afin de décider quel algorithme appliquer. Ce qui n'est pas chose aisée en pratique de part la présence des constantes dans les complexités. Mais, en réalité, il peut arriver que la densité des hits varie en fonction des régions des séquences étudiées. Cela suggère que pour les régions de forte densité en hits, il serait plus efficace d'utiliser l'algorithme de programmation dynamique alors que dans les régions de faible densité il serait plus

efficace d'appliquer l'algorithme par balayage. Nous verrons qu'il est possible d'établir un algorithme hybride qui permet de calculer une chaîne optimale en traitant les hits dans leur ordre d'apparition avec alternativement chacun des deux algorithmes et selon la densité en hits à la position étudiée.

Les problèmes généraux d'annotation, de classification ou de clusterisation des séquences ARN constituent un des problèmes majeurs de la bioinformatique et reposent sur la résolution efficace de la question suivante : étant données une séquence ARN requête et une base de séquences ARN cibles, quels sont les membres de la base dont la similarité avec la requête est suffisante pour indiquer une relation possible d'un point de vue évolutionnaire ou fonctionnel ? Cette question est depuis longtemps étudiée en ce qui concerne les séquences géniques codantes, cependant les molécules ARN présentent un défi puisque leurs fonctions ne reposent pas uniquement sur leur séquence mais également sur la conformation spatiale qu'elles adoptent (appelée *structure*). Alors que l'obtention de la structure tridimensionnelle est un problème extrêmement complexe, la prédiction d'une structure secondaire est un problème plus largement étudié (voir par exemple Lorenz et al. (2011)). Les structures secondaires d'ARN peuvent être modélisées, entre autres, par deux structures duales : les *séquences arc-annotées* et les *structures arborescentes*. Chacune de ces représentations présentent des caractéristiques propres expliquant leur utilisation différentielle suivant le type de problème à résoudre. De nombreuses méthodes permettent de comparer des séquences ARN deux à deux en prenant en compte la structure secondaire potentielle incluse dans ces séquences. La plupart des méthodes peuvent être classées en deux familles :

- les outils nécessitant la structure secondaire pour comparer les séquences, tels que *RNAforester* (Hochsmann et al., 2003; Schirmer and Giegerich, 2013), *Gardenia* (Blin et al., 2010) ou *MiGaL* (Allali and Sagot, 2008) pour n'en citer que trois (se référer à *Brasero* (Allali et al., 2008)).
- les outils ne nécessitant que des séquences ARN en entrée et utilisent un modèle de covariance ou une matrice de probabilité d'appariement des bases tels que *LocARNA* (Will et al., 2007), *Infernal* (Nawrocki and Eddy, 2013) ou *CARNAC* (Touzet and Perriquet, 2004).

La première famille d'approches repose sur les notions classiques de distance d'édition et d'alignements. Les ARN sont modélisés par des structures arborescentes ou des séquences arc-annotées et l'algorithme recherche soit à établir les opérations d'édition de coût minimal permettant de transformer le premier ARN en le deuxième, soit à maximiser le score d'alignement. Ainsi l'algorithme d'alignement de structures secondaires d'ARN sans pseudo-noeud le plus efficient est au moins cubique (Zhong and Zhang, 2013). Une telle complexité constitue la référence actuelle pour la comparaison deux à deux de structures, ce qui soulève la question de l'utilisation de telles approches lorsqu'un grand nombre de comparaisons sont requises, comme par exemple lors d'une étude de regroupement par similarité. La seconde famille de méthodes de comparaison d'ARN deux à deux utilise directement les séquences ARN. La méthode actuelle de référence, *LocARNA* (Heyne et al.,

2009), calcule l'alignement des séquences ARN à partir de la fonction de partition de l'ensemble des appariements possibles dans chacune des séquences, sous l'hypothèse d'une distribution de Boltzmann des énergies libres. *LocARNA* peut être utilisé pour réaliser des alignements multiples de séquences ARN mais peut aussi être limité à une comparaison deux à deux. Cependant dans les deux cas sa complexité demeure inchangée. Des filtres tels que *ExpARNA-P*/*Exploc-P* (Schmiedl et al., 2012) ont été implémentés et améliorent la vitesse d'alignement mais au détriment de l'optimalité des résultats. Ces méthodes reposent sur la conservation de motifs en séquence et en structure, appelés *EPM* (« Exact Pattern Matching »), qui peuvent être détectés en un temps quadratique, et sont fournis comme contraintes à *LocARNA*. Cela permet de diviser le calcul de l'alignement en plusieurs problèmes indépendants plus petits et ainsi de réduire le temps de calcul total.

Enfin un dernier type de méthodes permet de résoudre le problème de classification en assignant chaque ARN requête à une famille parmi un ensemble de familles prédéfinies, comme celle de la Rfam (Burge and al., 2013) par exemple. L'outil de classification de la Rfam, *Infernal* (Nawrocki and Eddy, 2013), calcule à partir des séquences appartenant à une même famille le modèle de covariance associé à cette famille. Ces modèles sont alors utilisés pour vérifier la similarité d'une séquence requête avec chacune des familles. Malgré de récentes améliorations, cette approche demeure gourmande en temps, ce qui a motivé le développement de filtres tels que *RNA sifter* (Janssen et al., 2008). Outre cette limitation en temps, *Infernal* est conçu pour les bases de données d'ARN connus et n'est en conséquence pas adapté au problème de regroupement *de novo* (ou « clustering ») par exemple.

Les travaux présentés dans ce manuscrit portent sur le problème de la comparaison d'un ARN requête Q et d'un ensemble D, potentiellement grand, de séquences d'ARN. Pour cela nous introduisons une nouvelle méthode, *RNA-unchained*, qui calcule de manière efficace des alignements de haute qualité entre Q et les membres de D. Notre méthode est basée sur le principe classique de comparaison de séquences et se décompose comme suit :

- Indexation des graines de chacune des séquences cibles (étape de pré-calcul réalisée une seule et unique fois).
- Recherche des hits, c'est-à-dire des graines de la séquence requête présentes dans l'index pré-calculé.
- Chaînage des hits entre la séquence requête et chacune des séquences cibles en une chaîne valide.
- Alignement exact contraint par la chaîne valide.

Notre méthode combine à la fois une étape d'indexation efficace de graines, des motifs en séquence et en structure, pour un jeu de séquences cibles fixé, et un algorithme d'alignement en séquence et structure efficace contraint par une ancre (ici *LocARNA*). De manière similaire à *LocARNA* et aux méthodes assimilées, *RNA-unchained* ne nécessite pas un ensemble de séquences cibles pré-organisées en familles.

Afin d'évaluer *RNA-unchained*, l'approche présentée par Schmiedl et al. (2012)

a été suivie. Le benchmark *BRAliBase2.1* (Wilm et al., 2006), qui est composé d'un ensemble de séquences d'ARNnc alignées par paires, a été utilisé comme jeu de données d'évaluation. Avec notre filtre, on observe des alignements de qualité comparable voire meilleure que *LocARNA*, et sensiblement meilleur que *ExpLoc-P*, pour des temps de calculs comparables entre *RNA-unchained* et *ExpLoc-P*.

Un second axe s'articule autour d'opérations de fouille de données, d'analyse et de comparaisons intra- et interspécifique des séquences génomiques, centrées sur l'extraction des familles de pseudogènes et la formalisation des règles régissant l'architecture du pseudome dans l'objectif d'automatiser et d'améliorer la prédiction de cette famille d'objets non-codants. L'ensemble de cette approche est formalisée par le développement d'une méthode d'identification et d'analyse topologique et phylogénétique des pseudogènes à partir de l'analyse de souches bactériennes du genre *Oenococcus* présentes au cours de la vinification.

La vinification est caractérisée par deux fermentations, qui doivent être maîtrisées par l'homme, impliquant la flore microbienne naturellement présente dans le moût de raisin. Tout d'abord les levures assurent la fermentation alcoolique, puis les bactéries lactiques réalisent la fermentation malolactique. La fermentation malolactique est un processus bénéfique et nécessaire à la production de la plupart des vins mais dépend de la capacité des bactéries lactiques autochtones à survivre et à se développer dans ces conditions physico-chimiques particulièrement hostiles. Dans ces conditions, les bactéries lactiques les plus résistantes, le plus souvent *Oenococcus oeni*, sont naturellement sélectionnées.

Oenococcus oeni apparaît comme « la bactérie la mieux adaptée aux conditions de vinification ». Cependant, une grande diversité de phénotypes (tolérance au vin, efficacité fermentaire,...) est naturellement observée au niveau des souches. En conséquence, le recours à l'inoculation avec des souches d'*Oenococcus oeni* sélectionnées est devenue pratique courante.

À ce jour, les génomes des différentes souches identifiées de l'espèce sont en cours de séquençage ou d'annotation. En particulier, les génomes de deux souches aux performances oenologiques antagonistes ont été entièrement séquencés et circularisés : la souche PSU-1 et la souche BAA-1163. En parallèle, les logiciels d'inférence de présence de gènes permettent l'automatisation de la cartographie génique. Cependant de nouvelles entités génomiques, les pseudogènes faussent ces prédictions automatiques.

La plasticité des génomes est connue pour être un point clef de l'évolution des génomes. Les modes de vie des bactéries peuvent mener à cette variabilité du génome. Son étude est une étape importante pour la compréhension de l'évolution et de l'adaptation des génomes à une niche écologique (Makarova and Koonin, 2007).

Les pseudogènes sont des séquences géniques dérivant de séquences géniques ayant codé pour des protéines qui ont perdu leur capacité suite à des altérations. Ils proviennent de divers mécanismes génétiques et évolutionnaires qui altèrent la transcription ou la traduction de ces séquences (Rouchka and Cha, 2009).

L'annotation des pseudogènes pour des génomes complets a été réalisée pour

plusieurs procaryotes (Liu et al., 2004; Lerat and Ochman, 2004). Les génomes du genre *Oenococcus* dérivent de la branche des *Leuconostoc* parmi les bactéries lactiques et constituent des génomes idéaux pour l'étude des variations du contenu des génomes. Premièrement, le genre *Oenococcus* se compose de deux espèces qui sont présentes uniquement dans certaines niches écologiques restreintes et les souches de ce genre sont les seules parmi les bactéries lactiques à avoir perdu leur capacité à réparer les altérations ponctuelles de leur génome.

Nous avons étudié le pseudome des génomes bactériens du vin afin de reconstruire et de modéliser l'histoire évolutive des séquences géniques. Nos travaux se focalisent sur les génomes du genre *Oenococcus* qui présentent un intérêt économique. De plus, ces souches présentent un génome hautement compact et des séquences pseudogéniques. Enfin, ces souches présentent de fortes variations génotypiques (Borneman et al., 2010; Bon et al., 2009) et des variations phénotypiques hétérogènes (Bilhère et al., 2009; Bartowsky and Borneman, 2011), ce qui rend ces espèces bactériennes hautement spécialisées difficiles à « domestiquer ».

La plupart des méthodes de détection reposent sur la comparaison de régions génomiques avec des séquences protéiques connues afin d'identifier de nouveaux pseudogènes. Cependant, cette approche ne permet pas la détection des pseudogènes unitaires pour lesquels il est possible qu'aucun gène de référence n'existe (Rouchka and Cha, 2009). Afin de déterminer la proportion de pseudogènes, provenant à la fois des processus d'évolution et des erreurs d'annotations, dans les génomes d'*Oenococcus oeni*, nous avons développé une méthode, *PseudOE*. Cette méthode permet de localiser les pseudogènes non identifiés ou mal identifiés au cours d'une annotation. Pour cela une analyse comparative des éléments des génomes des souches PSU-1 (Mills et al., 2005) et BAA-1163 (version complète non publique) est réalisée. Leur sélection en tant que jeu de données « test » présente un intérêt particulier du point de vue biotechnologique et phylogénétique puisque chacune des souches possède des propriétés phénotypiques distinctes (contrairement à BAA1163, PSU1 est un initiateur commercialisé) et appartient à un sous-groupe phylogénétique d'*Oenococcus oeni* différent (Bilhère et al., 2009).

L'analyse des deux souches d'*Oenococcus oeni* avec la méthode établie a permis de mettre en évidence des erreurs d'annotation de gènes codants, en particulier la présence de gènes présentant un « frameshift », mais aussi d'identifier au sein de régions non-codantes de nouvelles séquences pseudogénisées. Non seulement cette analyse des séquences nous a permis d'extraire de nouvelles données géniques des génomes, mais elle nous a aussi fourni des données quant aux processus d'évolution des gènes et d'adaptation de ces souches. Cette analyse a également permis d'observer la topologie de la répartition chromosomique de ces séquences au sein de ces souches particulières. Ainsi, suite à une étude comparative de l'ensemble des résultats obtenus, montrant aussi bien l'existence de pseudogènes spécifiques à une souche que la présence de séquences pseudogéniques conservées entre les souches, il semble raisonnable de penser que les pseudogènes forment un réservoir des possibilités évolutives dans lequel les souches puisent leur capacité d'adaptation à leur

environnement.

Ce manuscrit se compose de quatre chapitres organisés de manière à décrire l'ensemble des éléments nécessaires à la compréhension des filtres développés. Le premier décrit l'ensemble des informations nécessaires pour appréhender les concepts biologiques dans leur globalité, c'est-à-dire le contexte biologique à l'origine des objets, ARN et pseudogènes, étudiés. Il a pour objectif de permettre la compréhension des choix opérés au cours des travaux réalisés. Pour cela nous commencerons par les processus géniques permettant le décodage de l'information génétique contenu dans la molécule d'ADN et la synthèse des médiateurs des fonctions biologiques. Dans ce chapitre, nous détaillerons les deux objets géniques qui sont au coeur de la thèse : la molécule d'ARN et les séquences pseudogéniques.

Dans le deuxième chapitre nous nous intéressons à la comparaison des objets biologiques et nous présentons l'ensemble des fondements algorithmiques qui soutiennent les travaux réalisés dans la suite de ce manuscrit. Nous commencerons par l'étude des comparaisons de séquences et plus particulièrement par l'analyse des algorithmes d'alignement local et global. Nous poursuivrons par la présentation des algorithmes de chaînage au sein d'une séquence et entre deux séquences. Ce qui nous amènera à présenter un nouvel algorithme hybride de chaînage 2D. Enfin, nous terminerons par la description de deux filtres très utilisés pour la comparaison de séquences et dont la composition est similaire à celle du filtre que nous proposons dans le chapitre suivant.

Dans le troisième chapitre nous présenterons une nouvelle méthode : *RNA-unchained*. Dans un premier temps nous verrons qu'il existe plusieurs méthodes permettant d'intégrer la structure secondaire des séquences ARN lors de leur comparaison. Tout comme pour les séquences, il est possible d'utiliser des méthodes de chaînage tenant compte à la fois de la séquence mais aussi de la structure. Le filtre que nous présentons dans ce chapitre tire parti d'un tel algorithme, tout en incluant en amont et en aval un affinage des données.

Dans le dernier chapitre nous établirons une méthode permettant d'identifier et d'analyser les relations phylogénétiques des pseudogènes au sein d'un ou plusieurs génomes. Pour cela nous commencerons par établir une nomenclature des objets géniques. Nous détaillerons alors les différentes étapes permettant d'identifier, de caractériser puis de classifier les pseudogènes par comparaison de séquences. Nous terminerons par une analyse comparée du pseudome de deux génomes de l'espèce *Oenococcus oeni*.

Nous conclurons par les perspectives envisagées suite à la réalisation de ces travaux, concernant les améliorations techniques envisageables pour chacun des filtres, les applications dans des domaines biologiques élargis et pour terminer l'utilisation conjointe des filtres sur une même problématique. Il serait, en effet, alors intéressant d'analyser le degré de conservation de la structure secondaire des objets géniques codants et non codants. Pour cela, l'utilisation du filtre RNA-unchained développé pourrait apporter des réponses quant à la conservation de la structure secondaire des ARN de deux objets homologues, l'un codant et l'autre non. La structure secondaire

des ARN non codants des séquences pseudogéniques pourraient alors contribuer à l'identification de tels objets au sein des génomes.

Bibliographie

Allali, J., d'Aubenton Carafa, Y., Chauve, C., Denise, A., Drevet, C., Ferraro, P., Gautheret, D., Herrbach, C., Leclerc, F., De Monte, A., et al. (2008). Benchmarking rna secondary structure comparison algorithms. *Actes des Journées Ouvertes de Biologie, Informatique et Mathématiques-JOBIM'08*, pages 67–68.

Allali, J. and Sagot, M. (2008). A multiple layer model to compare rna secondary structures. *Software : Practice and Experience*, 38(8) :775–792.

Bartowsky, E. and Borneman, A. (2011). Genomic variations of oenococcus oeni strains and the potential to impact on malolactic fermentation and aroma compounds in wine. *Applied microbiology and biotechnology*, 92(3) :441–447.

Bilhère, E., Lucas, P., Claisse, O., and Lonvaud-Funel, A. (2009). Multilocus sequence typing of oenococcus oeni : detection of two subpopulations shaped by intergenic recombination. *Applied and environmental microbiology*, 75(5) :1291–1300.

Blin, G., Denise, A., Dulucq, S., Herrbach, C., and Touzet, H. (2010). Alignments of rna structures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(2) :309–322.

Bon, E., Delaherche, A., Bilhere, E., De Daruvar, A., Lonvaud-Funel, A., and Le Marrec, C. (2009). Oenococcus oeni genome plasticity is associated with fitness. *Applied and environmental microbiology*, 75(7) :2079–2090.

Borneman, A., Bartowsky, E., McCarthy, J., and Chambers, P. (2010). Genotypic diversity in oenococcus oeni by high-density microarray comparative genome hybridization and whole genome sequencing. *Applied microbiology and biotechnology*, 86(2) :681–691.

Burge, S. and al. (2013). Rfam 11.0 : 10 years of rna families. *Nucleic Acids Research*, pages D226–D232.

Cech, T., Bennett, D., Jasny, B., Kelner, K., Miller, L., Szuromi, P., Voss, D., Kiberstis, P., Parks, S., Ray, L., et al. (1992). The molecule of the year. *Science*, 258 :1861.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences : computer science and computational biology*. Cambridge University Press.

Heyne, S., Will, S., Beckstette, M., and Backofen, R. (2009). Lightweight comparison of rnas based on exact sequence-structure matches. *Bioinformatics*, page btp065.

Hochsmann, M., Toller, T., Giegerich, R., and Kurtz, S. (2003). Local similarity in rna secondary structures. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 159–168. IEEE.

Höhl, M., Kurtz, S., and Ohlebusch, E. (2002). Efficient multiple genome alignment. *Bioinformatics*, 18(suppl 1) :S312–S320.

Janssen, S., Reeder, J., and Giegerich, R. (2008). Shape based indexing for faster search of rna family databases. *BMC bioinformatics*, 9(1) :131.

Kertesz, M., Wan, Y., Mazor, E., Rinn, J., Nutter, R., Chang, H., and Segal, E. (2010). Genome-wide measurement of rna secondary structure in yeast. *Nature*, 467(7311) :103–107.

Lerat, E. and Ochman, H. (2004). ψ-ϕ : Exploring the outer limits of bacterial pseudogenes. *Genome research*, 14(11) :2273–2278.

Lipman, D. and Pearson, W. (1985). Rapid and sensitive protein similarity searches. *Science*, 227(4693) :1435–1441.

Liu, Y., Harrison, P., Kunin, V., and Gerstein, M. (2004). Comprehensive analysis of pseudogenes in prokaryotes : widespread gene decay and failure of putative horizontally transferred genes. *Genome biology*, 5(9) :R64.

Lorenz, R., Bernhart, S., Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P., Hofacker, I., et al. (2011). Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1) :26.

Makarova, K. and Koonin, E. (2007). Evolutionary genomics of lactic acid bacteria. *Journal of bacteriology*, 189(4) :1199–1208.

Mills, D., Rawsthorne, H., Parker, C., Tamir, D., and Makarova, K. (2005). Genomic analysis of oenococcus oeni psu-1 and its relevance to winemaking. *FEMS microbiology reviews*, 29(3) :465–475.

Nawrocki, E. and Eddy, S. (2013). Infernal 1.1 :100-fold faster rna homology searches. *Bioinformatics*, 29(22) :2933–2935.

Ohlebusch, E. and Abouelhoda, M. I. (2006). Chaining algorithms and applications in comparative genomics. *Handbook of Computational Molecular Biology*.

Rouchka, E. and Cha, I. (2009). Current trends in pseudogene detection and characterization. *Current Bioinformatics*, 4(2) :112–119.

Schirmer, S. and Giegerich, R. (2013). Forest alignment with affine gaps and anchors, applied in rna structure comparison. *Theoretical Computer Science*, 483 :51–67.

Schmiedl, C., Möhl, M., Heyne, S., Amit, M., Landau, G., Will, S., and Backofen, R. (2012). Exact pattern matching for rna structure ensembles. In *Research in Computational Molecular Biology*, pages 245–260. Springer.

Touzet, H. and Perriquet, O. (2004). Carnac : folding families of related rnas. *Nucleic acids research*, 32(suppl 2) :W142–W145.

Wan, Y., Kertesz, M., Spitale, R., Segal, E., and Chang, H. (2011). Understanding the transcriptome through rna structure. *Nature Reviews Genetics*, 12(9) :641–655.

Will, S., Reiche, K., Hofacker, I., Stadler, P., and Backofen, R. (2007). Inferring noncoding rna families and classes by means of genome-scale structure-based clustering. *PLoS computational biology*, 3(4) :e65.

Wilm, A., Mainz, I., and Steger, G. (2006). An enhanced rna alignment benchmark for sequence alignment programs. *Algorithms for molecular biology*, 1(1) :19.

Zhong, C. and Zhang, S. (2013). Efficient alignment of rna secondary structures using sparse dynamic programming. *BMC bioinformatics*, 14(1) :269.

Chapitre 1

Contexte Biologique

Au cours des travaux qui vont être exposés dans ce manuscrit, un certain nombre de termes biologiques vont être employés. Afin d'appréhender au mieux l'ensemble des notions abordées, il paraît utile de les définir. C'est le but de ce premier chapitre. De plus, une simple définition n'est pas suffisante, c'est pourquoi l'ensemble de ces éléments seront définis et remis dans leur contexte biologique. Pour cela, une vision globale de la cellule et de son fonctionnement d'un point de vue génétique sera donné. Puis cette étude se focalisera sur deux objets biologiques : les ARN et les pseudogènes. Enfin ce premier chapitre s'achèvera par une description des méthodes de comparaisons d'objets biologiques sur lesquelles sont basés l'ensemble de nos travaux.

1.1 La Cellule : Unité du Vivant

1.1.1 Généralités sur la Cellule

En 1665 Robert Hooke observe pour la première fois dans du liège des structures lui rappelant les cellules des monastères (Dufey, 1986). C'est pour cette raison qu'il décide de nommer ces objets cellules. Mais ce n'est qu'en 1839 que Theodor Schwann met en évidence que la cellule est l'unité commune de structure et de développement de l'ensemble des organismes en observant par microscopie que les organismes tels que les plantes et les animaux sont tous composés de cellules (Schwann, 1837). Ces observations sont à la base de la théorie cellulaire.

(i) La Théorie Cellulaire

La théorie cellulaire repose sur cinq postulats :
- Tout être vivant est constitué d'une ou plusieurs cellules.
- Toute cellule provient d'une autre cellule (division cellulaire).
- Unité vivante et unité de base du vivant : unité autonome capable de réaliser les fonctions nécessaires et suffisantes à la vie.

- Individualité cellulaire garantie par la membrane plasmique qui régule les échanges entre la cellule et son environnement.
- Elle renferme l'information nécessaire à son fonctionnement et à sa reproduction sous forme d'Acide DésoxyriboNucléique (ADN) [1].

Ainsi la cellule est définie comme l'unité structurelle, fonctionnelle et reproductrice du vivant.

(ii) Les Types Cellulaires

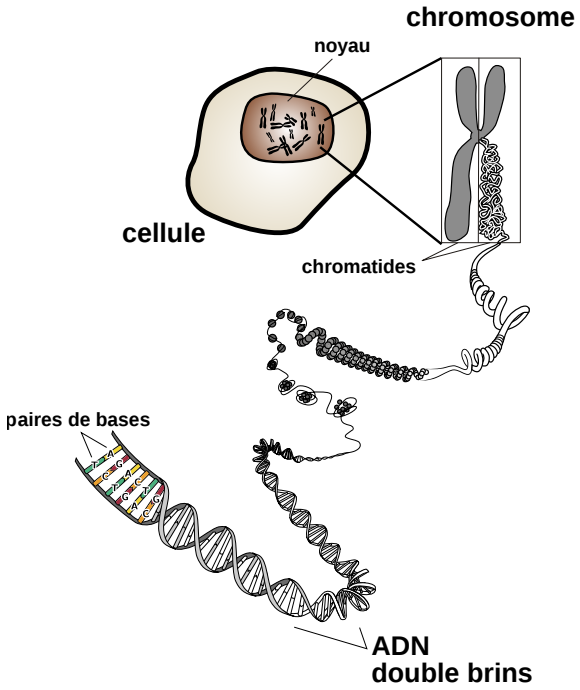


FIGURE 1.1 – *Compaction de la molécule d'ADN en entités chromosomiques regroupées dans un noyau cellulaire.*[2]

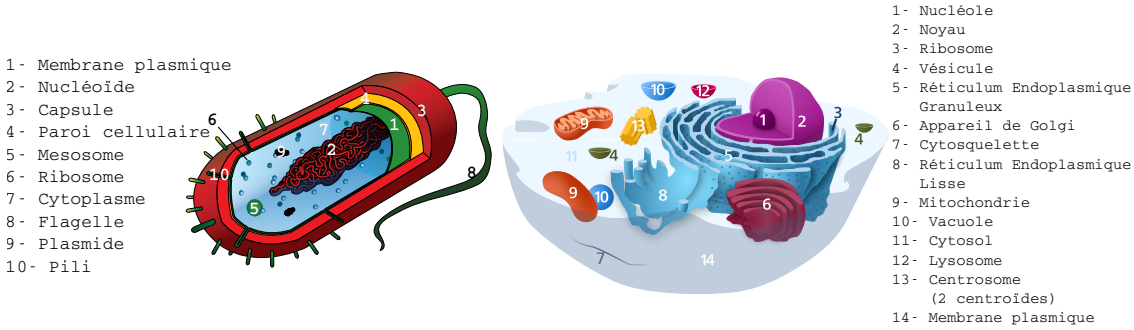
La molécule d'ADN contenue dans les cellules peut être regroupée à l'aide de protéines en une structure particulière, les chromosomes (voir la Figure 1.1). Ces chromosomes se retrouvent au sein d'une cellule sous deux formes :

- une forme libre, présente dans le cytoplasme de la cellule.
- une forme confinée, présente dans le noyau cellulaire ou dans certains organites, la mitochondrie et le chloroplaste [3].

Ces deux modes de stockage de l'information génétique caractérisent les deux types cellulaires existants :

1. La nature des virus comme appartenant au monde du vivant ou de l'inerte reste encore à ce jour une question ouverte. Il est tout de même intéressant de noter l'existence de virus à ARN (Acide RiboNucléique) en outre des virus à ADN.
2. Illustration de Cnickelfr.
3. Organite spécifique aux cellules végétales

- Les cellules *procaryotes* dont l'ADN est sous forme libre dans le cytoplasme cellulaire.
- Les cellules *eucaryotes* dont l'ADN est sous forme confinée dans le noyau cellulaire [4].



(a) cellule procaryote [5] *(b) cellule eucaryote* [6]

Figure *1.2 – Organisations des cellules eucaryotes et procaryotes*

En plus de la localisation différente de l'information génétique, on observe une organisation différente entre ces deux types cellulaires (voir Figure 1.2). Les cellules procaryotes ne présentent pas d'organisation en compartiments (voir Figure 1.2.a), l'ensemble des fonctions cellulaires est assuré au sein du cytoplasme. Par comparaison, les cellules eucaryotes présentent une forte compartimentation (voir Figure 1.2.b). Chacun de ces compartiments est alors le siège d'une ou plusieurs activités cellulaires, comme la fonction de respiration dans la mitochondrie ou la synthèse protéique dans le réticulum endoplasmique granuleux pour ne citer que ces deux exemples. L'ensemble de ces compartiments coopère afin de réaliser l'ensemble des fonctions cellulaires.

(iii) Les Organismes Cellulaires

La cellule en tant qu'unité de base du vivant constitue un organisme et on distingue deux types d'organismes :

- Les organismes unicellulaires qui se composent d'une unique cellule. On retrouve parmi les unicellulaires l'ensemble des procaryotes comme par exemple les bactéries et certains eucaryotes, les levures.
- Les organismes pluricellulaires qui se composent de plusieurs cellules. On retrouve parmi les pluricellulaires l'ensemble des organismes dits « complexes »,

4. On notera qu'au cours de certaines phases du cycle cellulaire l'ADN peut également être présent sous forme libre dans le cytoplasme des cellules eucaryotes.
5. Illustration de Mariana Ruiz Villarreal.
6. Illustration de Kelvinsong.

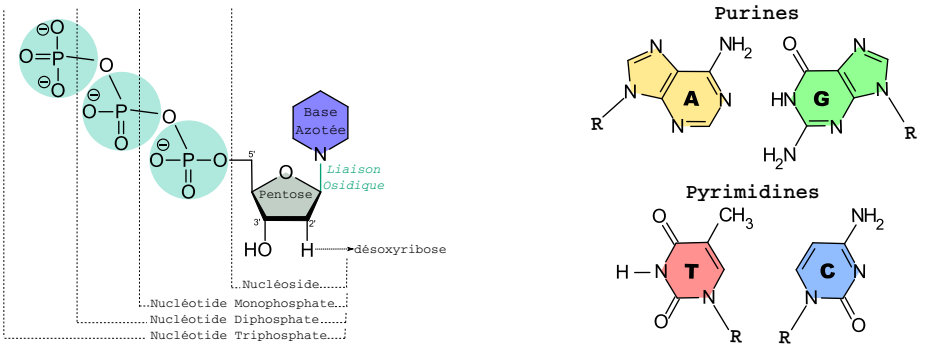
de part leur organisation cellulaire, comme par exemple l'Homme constitué de 10^{14} cellules.

Chaque cellule composant un organisme pluricellulaire fonctionne de manière autonome tout en étant coordonnée avec d'autres cellules, souvent colocalisées, et spécialisées dans une fonction partagée. On dit que ces cellules sont différenciées. Chaque ensemble de cellules différenciées dans un même type constitue un tissu cellulaire. La combinaison de différents tissus cellulaires impliqués dans une fonction commune forme alors un organe.

Chacune des cellules d'un organisme possède la totalité de l'information génétique permettant à cet organisme de réaliser les activités du vivant de la respiration à la production d'énergie. De plus, la cellule qui est alors la plus petite unité constitutive du vivant, est aussi une unité spécialisée grâce à l'expression différentielle de cette information génétique codée dans son ADN. L'ensemble de cette information est appelé génome. Toute cellule présente effectivement un fonctionnement de base correspondant aux activités indispensables à sa survie, comme la respiration pour n'en citer qu'une. Toute cellule spécialisée d'un organisme pluricellulaire, de part sa différenciation, présente en plus de son activité de base une activité spécifique liée à sa fonction au sein du tissu, de l'organe, qu'elle compose. Il est ici intéressant de noter que, suivant son environnement et l'âge de l'organisme qu'elle compose, la cellule adapte son fonctionnement aux contraintes qu'elle subit.

1.1.2 Les Processus d'Expression de l'Information Génétique

(i) De l'ADN ... à l'ARN



(a) Composition d'un désoxyribonucléotide. *(b) Les quatre bases azotées de l'ADN.*

FIGURE *1.3 – Composition des quatre nucléotides de la molécule d'ADN.*

La molécule d'ADN La molécule d'ADN peut être décrite comme un collier de perles de quatre couleurs différentes. En effet, cette molécule d'ADN est constituée par la succession de quatre désoxyribonucléotides (plus couramment appelés *nucléotides*) universels pouvant être assimilés à des perles. Un nucléotide est lui-même une molécule organique composée d'un sucre, le pentose, d'un groupement phosphate et d'une base azotée (voir Figure 1.3). Le groupement phosphate et le pentose étant commun à tout nucléotide, la nature de ce dernier est déterminée par la base azotée qu'il porte. Les couleurs des perles font alors référence aux quatre différentes bases azotées pouvant constituer les nucléotides de l'ADN : l'adénine (A), la cytosine (C), la guanine (G) et la thymine (T). En outre, les nucléotides qui sont des molécules chimiques carbonées ont donc leurs cinq carbones numérotés de 1 à 5. On remarque sur la Figure 1.3 que le groupe hydroxyde est numéroté $3'$ et que le groupe phosphate est numéroté $5'$. Or ces deux groupes caractéristiques sont ceux qui entrent en réaction pour lier entre elles les perles de nucléotides et constitue le fil de ce collier. Ainsi dans l'ADN, les nucléotides sont reliés grâce à des liaisons $3' - 5'$ phosphodiester. On parle alors d'orientation $3' \rightarrow 5'$ de l'ADN [7]. En résumé, pour constituer un brin d'ADN, il suffit d'enchainer selon une certaine séquence des nucléotides grâce à ces liaisons fortes (voir Figure 1.4).

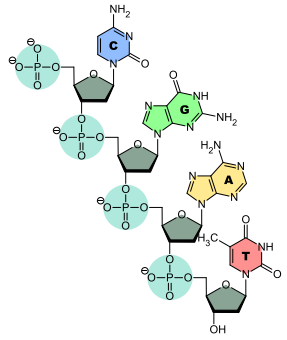


Figure 1.4 – *Brin d'ADN formé par la succession des nucléotides A, C, G et T.*[8]

Cette molécule concentrant toute l'information génétique d'un individu n'est pas qu'un simple collier de perles. En effet, le 25 avril 1953 un généticien, James Watson et un physicien, Francis Crick, publient dans *Nature* la première modélisation de la structure de l'ADN sur la base des travaux de Rosalind (Crick and Watson, 1953). Une découverte qui sera saluée par l'obtention d'un prix Nobel en 1962. Tout d'abord, au cours du paragraphe précédent il a été explicité que l'enfilage des perles nucléotidiques est réalisé grâce à une liaison forte, une liaison phosphodiester, entre le phosphate d'un nucléotide et le sucre, le désoxyribose, du nucléotide suivant. De plus, quelques années auparavant, il a été démontré que, d'une espèce à une autre, le pourcentage de chaque nucléotide varie mais que les rapports $\frac{A}{T}$ et $\frac{C}{G}$ restent proches de 1. On observe par exemple chez l'Homme : $A = 30,4\%$; T=30,1% ; C=19,6% et G=19,9%. Ce qui a permis la conjecture de la complémentarité des bases $A - T$ et $C - G$. Cette hypothèse ainsi qu'un cliché de l'ADN obtenu par diffraction au rayon X (voir Figure 1.5) ont alors permis à Crick et Watson de définir la structure en double hélice de l'ADN (Crick and Watson, 1953).

7. Cette orientation est celle du sens de lecture des brins au cours de la traduction.
8. D'après une illustration de Sponk

La double hélice décrite par Watson et Crick se compose donc de deux brins d'ADN. Cet appariement des deux brins est rendu possible par la *complémentarité* des nucléotides qui les composent. En effet, les bases azotées de chaque brin établissent des liaisons faibles permettant de relier les deux brins entre eux. Ces liaisons faibles sont des liaisons hydrogènes établies entre deux bases azotées : on compte deux liaisons entre A et T et trois liaisons entre C et G. Ainsi dans la double hélice si on trouve un A (réciproquement un C) sur l'un des brins on trouve en face sur le second brin, ou *brin complémentaire*, un T (réciproquement un G) et inversement (voir Figure 1.6). Les paires de bases A-T (réciproquement C-G) sont appelées paires de bases complémentaires.

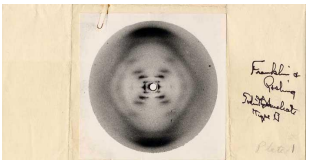


FIGURE 1.5 – *Diffraction aux rayons X de la molécule d'ADN obtenue par Rosalind Franklin en 1952 et ayant permis à Crick et Watson de révéler la structure en double hélice de l'ADN.*

On peut définir une molécule d'ADN comme une hélice (voir Figure 1.7) de deux séquences polynucléotidiques complémentaires et *antiparallèles*. On définit par antiparallèle deux brins de polarité inverse. Ainsi l'un des deux brins est orienté dans le sens $5' \rightarrow 3'$ alors que le brin complémentaire est orienté dans le sens $3' \rightarrow 5'$. Il est donc très facile de retrouver le brin complémentaire d'une séquence d'ADN donnée.

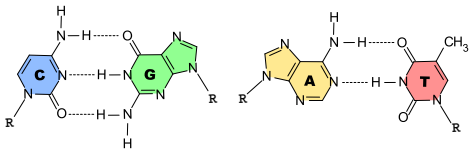


FIGURE 1.6 – *Représentation des liaisons hydrogènes établies entre les bases complémentaires de l'ADN et permettant l'établissement de la structure en double hélice des deux brins antiparallèles de l'ADN.*

Cette molécule d'ADN se retrouve sous deux formes (voir Figure 1.1) :

- Une molécule d'ADN linéaire présente chez les eucaryotes dans leur noyau, on parle d'ADN nucléaire linéaire.
- Une molécule d'ADN circulaire commune aux procaryotes que l'on retrouve également dans la mitochondrie et le chloroplaste.

Cette molécule d'ADN se replie sur elle-même pour former une nouvelle structure, le chromosome. Pour l'ADN circulaire on parle de chromosome circulaire alors que l'ADN linéaire se replie généralement en plusieurs chromosomes. On note que ce repliement dans l'espace des chromosomes permet de délimiter des territoires chromosomiques.

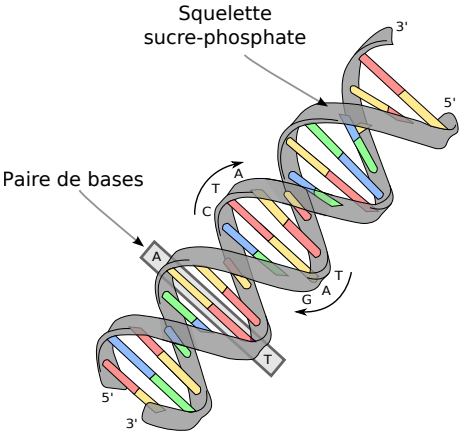


Figure 1.7 – *Structure en double hélice de l'ADN. On observe ici l'appariement des bases complémentaires A-T et C-G.*[9]

La molécule d'ADN est mesurée en paires de bases (noté *bp*) par référence aux paires formée par les deux brins de sa structure ou encore en nombre de bases (en kilobase *kb* ou mégabase *Mb*). La taille du génome varie d'un individu à un autre, indépendamment de la taille de l'individu et sans corrélation avec le niveau d'organisation de cet individu. Le pin à torches, *Pinus taeda*, possède l'un des plus longs génomes séquencés à ce jour avec ses 23 milliards de paires de bases, soit environ sept fois la longueur du génome humain qui possède déjà 3 milliards de paires de bases. La taille de la molécule d'ADN varie de quelques milliers de paires de bases à plusieurs dizaines de milliards de paires de bases pour les molécules séquencées.

Le défi repose non seulement dans la capacité à collecter ces données génétiques, soit l'extraction et le traitement des séquences ADN, mais aussi sur la capacité à assembler dans le bon ordre ces séquences et enfin à interpréter ces séquences assemblées.

Dans la séquence d'ADN, l'alternance des quatre bases forme le génome de l'individu. Toutes ces séquences générées, portant l'information génétique, constituent un message codé décrypté par la cellule. En effet, l'ordre, la nature, et le nombre de nucléotides dans la séquence considérée déterminent l'information génétique. La molécule d'ADN se compose d'une succession de nucléotides et comprend différents types de séquences : des séquences codantes, c'est-à-dire des séquences qui codent pour des protéines actives, et des séquences non codantes (la section 1.1.3 aborde ces notions).

La transcription La transcription est un processus biologique qui consiste en la reproduction d'une portion de la molécule d'ADN en molécule d'ARN. Cette réaction de transcription est catalysée par une enzyme appelée ARN polymérase (ou ARN_P). Différentes ARN polymérases se partagent cette activité selon le type d'ARN à synthétiser : ARNm, ARNt... Cette enzyme se fixe en amont de la séquence à transcrire sur un motif particulier : le site promoteur.

Le brin d'ADN sur lequel la polymérase est fixée sert alors de matrice à la synthèse du brin d'ARN complémentaire. Dans le cas des ARNm, la molécule d'ARN synthétisée est alors traduite en protéine sous l'action combinée des ARNt et des ribosomes : c'est le mécanisme de traduction.

Même si le principe de la transcription est ubiquitaire il existe de nombreuses différences selon l'appartenance de la cellule à un organisme eucaryote ou à un

9. D'après une illustration de MesserWoland

organisme procaryote.

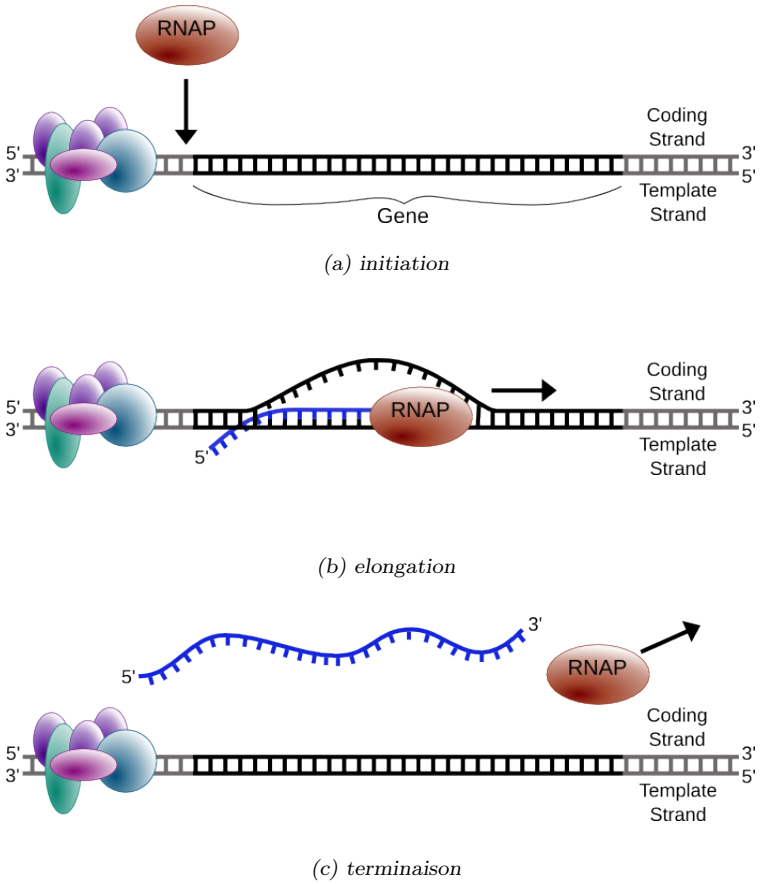


FIGURE 1.8 – *Les trois étapes successives de la transcription.*[10]

La transcription procaryote Chez les procaryotes dont les cellules ne présentent pas d'organisation complexe et où l'ADN se trouve dans le cytoplasme, la transcription se déroule donc également dans le cytoplasme cellulaire. Cette synthèse de l'ARN peut être décomposée en trois étapes.

(1) L'initiation (voir Figure 1.8.a) correspond à la fixation de l'ARN polymérase sur le site promoteur directement en amont de la séquence codante. La séquence consensus du promoteur, le plus souvent la *boite Pribnow* ($TATAAT$), est reconnue par l'une des sous-unités composant la polymérase.

(2) A l'aide d'un complexe protéique s'ensuit la phase d'élongation (voir Figure 1.8.b) au cours de laquelle le complexe enzymatique synthétise la molécule d'ARN complémentaire à celle de la matrice d'ADN.

10. D'après des illustrations de Forluvoft

(3) Enfin cette transcription prend fin lorsque la polymérase rencontre le terminateur situé directement après la séquence : c'est la troisième et dernière phase, la terminaison (voir Figure 1.8.c).

La molécule d'ARNm ainsi synthétisée est directement traduisible par la cellule. Chez les procaryotes, de par l'absence de compartimentation, la transcription et le processus consécutif, la traduction, peuvent se dérouler en parallèle sur une même séquence : il est possible d'observer la traduction d'une molécule d'ARNm dont la synthèse est toujours en cours.

La transcription eucaryote Par comparaison avec la transcription chez les procaryotes de multiples différences peuvent être relevées. Tout d'abord, l'ADN étant localisé dans le noyau chez les eucaryotes, la transcription a lieu dans cette organelle. De plus, l'ARN synthétisé n'est pas directement utilisable et doit subir la maturation post-transcriptionnelle. Alors que chez les procaryotes seule une ARN polymérase est à l'origine de la transcription, chez les eucaryotes trois types d'enzymes [11] sont répertoriés et interviennent séparément selon le type de séquence à transcrire. De plus, celles-ci ne sont pas suffisantes pour réaliser la synthèse d'ARN elles doivent s'associer à des facteurs de transcription avec lesquelles elles forment des complexes protéiques. Ces complexes réagissent selon les trois même phases que précédemment.

(1) Au cours de la phase d'initiation (voir Figure 1.8.a) le complexe reconnaît la « boite TATA » , environ 30 paires de bases en amont de la séquence génique à transcrire, sur laquelle elle se fixe. Deux autres séquences consensus, la « boite CAAT », modulatrice de l'expression, à -75 nucléotides environ, et la « boite GC », site de fixation, à -90 nucléotides environ.

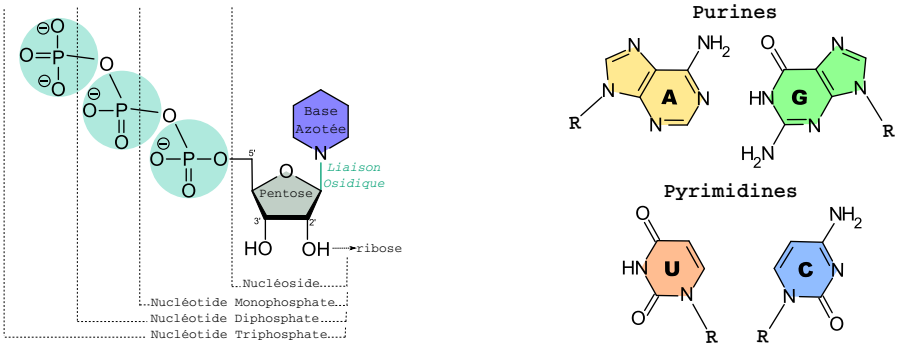
(2) Une fois le complexe de transcription fixé à la boite TATA, la deuxième phase, l'élongation (voir Figure 1.8.b), débute et ne s'arrête qu'à l'apparition du signal de polyadénylation AAUAAA.

(3) La transcription en elle-même est alors terminée (voir Figure 1.8.c) mais l'ARN obtenu, préARNm, n'est pas fonctionnel et nécessite encore trois étapes de maturation.

La molécule d'ARN La molécule d'ARN est constituée d'un enchaînement de ribonucléotides : l'adénine (A), la cytosine (C), la guanine (G) et l'uracile (U), reliés entre eux par des liaisons nucléotidiques (voir la Figure 1.9). L'ordre de ces nucléotides est dicté par la séquence des désoxyribonucléotides portés par la séquence ADN dont ils sont issus suite au processus de transcription.

Les ribonucléotides de l'ARN diffèrent des désoxynucléotides de l'ADN par la présence d'un groupement OH en $2'$ du ribose (et non d'un H comme le désoxyribose de l'ADN, voir Figure 1.9), mais aussi par le fait que la thymine (T) est substituée par l'uracile (U).

11. Un quatrième type d'ARN polymérase est répertorié et intervient au cours de la transcription de l'ADN mitochondrial.



(a) Composition d'un ribonucléotide. *(b) Les quatre bases azotées de l'ARN.*

FIGURE *1.9 – Composition des quatre nucléotides de la molécule d'ARN.*

À l'inverse de l'ADN, la plupart du temps structuré en double hélice, l'ARN peut adopter des conformations différentes (en simple brin, en tige boucle, ...) liées à sa fonction.

La molécule d'ARN étant à la base d'une partie des travaux exposés dans ce manuscrit, elle est l'objet d'une étude plus approfondie et la section 1.3.1 lui est dédiée.

Les modifications post-transcriptionnelles Les molécules d'ARN alors synthétisées sont soumises à des modifications visant, entre autres, à les stabiliser dans le milieu cellulaire ainsi qu'à les modifier en vue de la phase suivante.

Chez les procaryotes La molécule d'ARN synthétisée par l'ARNpolymérase ne nécessite pas de modification additionnelle chez les procaryotes.

Chez les eucaryotes *La maturation* Chez les eucaryotes, après un clivage du préARNm nouvellement synthétisé, au niveau du signal de polyadénylation, une nouvelle polymérase, la poly A polymérase ou PAP, ajoute plusieurs (jusqu'à 200 chez les eucaryotes supérieurs) résidus d'adénine à l'extrémité terminale 3' du pré-ARNm. Cette première modification permet d'assurer la stabilité de la molécule : on nomme cette élongation la *queue polyA*. Sur l'autre extrémité de la molécule, à l'extrémité 5', une coiffe méthylguanosine est nécessaire à la reconnaissance de l'ARN par les ribosomes. Une fois ces trois étapes terminées l'ARN obtenu, appelé pré-ARNm, n'est pas encore prêt pour l'étape suivante, la traduction. Il doit subir une dernière modification de maturation post-transcriptionnelle de par la conformation des séquences géniques chez les eucaryotes. *L'épissage* En effet les séquences codantes eucaryotes se composent elles mêmes de séquences codantes, les *exons*, et de séquences non codantes, les *introns*. Alors que les exons interviennent dans la

compositions de la protéine finale, les introns sont eux éliminés au cours d'une opération d'*épissage* par un complexe protéique particulier, le spliceosome. Au cours de cette étape d'excision une sélection des exons à conserver pour la traduction en protéine peut être opérée : on parle alors d'*épissage alternatif*. L'ARNm obtenu est alors plus court et quitte ensuite le noyau cellulaire pour le cytoplasme.

(ii) De l'ARN ... aux Fonctions Biologiques

Tout comme l'alphabet de l'ADN, celui de l'ARN se compose de quatre bases complémentaires deux à deux. Comme expliqué dans la section précédente il est facile de déduire la séquence d'ARN qui sera issue d'une séquence ADN en utilisant cette complémentarité des bases. L'étape suivant la transcription est la traduction. Au cours de ce processus la molécule d'ARN, composée des quatre bases de son alphabet, est traduite en une protéine, dont l'alphabet compte 22 acides aminés. Il existe donc une table de traduction de la composition en bases azotées de la molécule d'ARN en acides aminés protéiques : c'est le *code génétique*.

Le code génétique Ce code génétique est ainsi le lien entre le génotype (c'est-à-dire l'ensemble des gènes présents sur la molécule d'ADN) et les caractères de l'organisme (appelés phénotype). Ce génotype est déchiffrable grâce à un code commun à l'ensemble des êtres vivants : le code génétique (voir la Figure 1.10). Ce code génétique est partagé par l'ensemble des organismes, c'est pourquoi on définit l'ADN comme le support universel de l'information génétique.

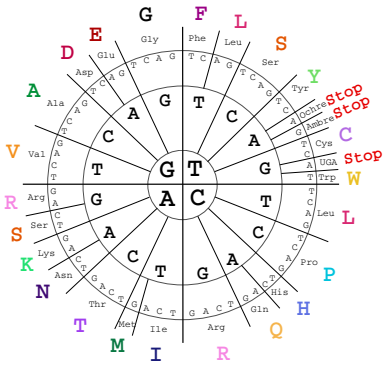


Figure 1.10 – *Code génétique le plus fréquemment utilisé pour la traduction.*

Le code génétique se lit triplet de bases par triplet de bases. Une suite de trois nucléotides est ainsi nommée codon. Chacun de ces codons est associé à un *acide aminé*. On dénombre 20 acides aminés [12] directement encodés dans la molécule d'ADN, une dizaine d'autres dérive des précédents par modification post-traductionnelle et enfin des dizaines d'autres n'entrent pas dans la composition des protéines. On peut rapprocher le décodage de l'ADN d'un langage dans lequel les codons de l'ADN constituent les mots. Cependant, ce langage déjà limité à 4^3 mots possibles contient aussi de la redondance. En effet, plusieurs codons peuvent coder pour un même mot (voir Figure 1.10), et donnent la dégénérescence du code génétique ou plus particulièrement de la troisième base du codon. En effet, dans la plupart des cas cette troisième base n'est pas significative et les deux premières suffisent à la traduction du codon.

12. 2 acide aminés spécifiques supplémentaires, sélénocysteine et pyrrolysine, peuvent être insérés spécifiquement au niveau de codons STOP.

Il est important de noter que des variantes de ce code génétique existent dans certaines lignées évolutives. En outre du code génétique standard on dénombre 17 autres codes génétiques parmi lesquels :

-2- Code génétique mitochondriale chez les vertébrés
-3- Code génétique mitochondriale chez les levures
-5- Code génétique mitochondriale chez les invertébrés
-11- Code génétique des bactéries, des archées et des plamides végétals
- ...

Ces variations se manifestent par une traduction différente d'un ou plusieurs codons. Le système de codage reste quant à lui inchangé.

Dans ce code certaines particularités ont également été remarquées telles que la traduction de deux manières différentes d'un même codon au sein d'une même espèce. Chez *Escherichia coli*, une ambiguité du code a été relevée puisque le codon UGA code tantôt pour le 21ième acide aminé, la sélénocystéine, tantôt pour un codon STOP.

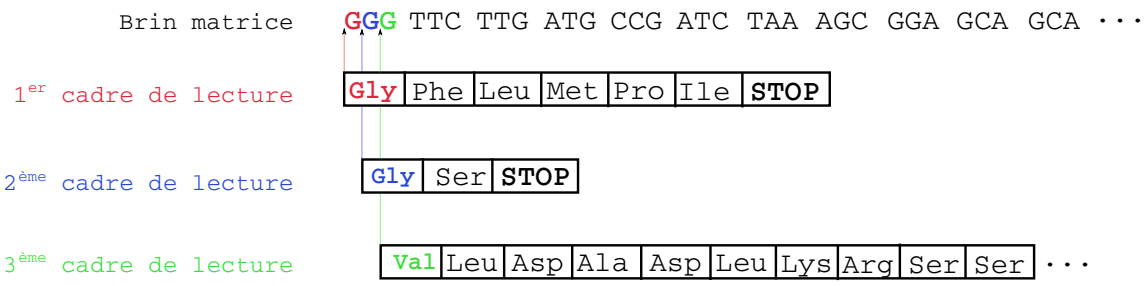


FIGURE 1.11 – *Phases de lecture. Selon le nucléotide pris comme premier constituant d'un codon, la traduction en acide aminé sera différente. Les deux brins de la molécule d'ADN pouvant servir de matrice on dénombre 6 cadres de lecture.*

Les phases de lecture On définit une phase, ou cadre, ouverte de lecture (notée *ORF* pour « Open Reading Frame ») par la région comprise entre deux codons STOP et présentant éventuellement un codon START. La présence dans une ORF d'un codon initiateur, tel que AUG peut permettre de retrouver la *CDS* (« Coding DNA Sequence ») qui se termine généralement par le codon STOP de l'ORF.

On remarque que chaque séquence ARN peut contenir trois phases ouvertes de lecture décalées d'un nucléotide par rapport aux autres (voir la Figure 1.11). De plus, sur l'ADN il peut y avoir transcription en ARN de chacun des deux brins, ce qui conduit à un total de six phases ouvertes de lecture.

La traduction L'ARNm présent dans le cytoplasme forme un complexe avec le ribosome qui peut être libre ou associé au réticulum endoplasmique (alors appelé Réticulum Endoplasmique Rugueux ou REG, voir Figure 1.2.b). Le ribosome est un complexe ribonucléoprotéique, formé d'ARNr (Noller et al., 1992) portant l'activité catalytique et de protéines ribosomiques. Il a pour fonction de décoder l'information

contenue dans l'ARNm pour synthétiser la protéine correspondante. Ce complexe est formé de deux sous-unités :

(1) une petite sous-unité chargée de l'interprétation des codons de la séquence ARN en acide aminés

(2) une grosse sous-unité chargée de la polymérisation des acides aminés selon l'ordre de décodage par la petite sous-unité pour former la protéine.

Cette traduction peut être amplifiée par l'association simultanée sur un même ARNm de plusieurs ribosomes. Ce chapelet de ribosomes consécutifs est nommé *polysome*.

La traduction se déroule alors en trois temps :

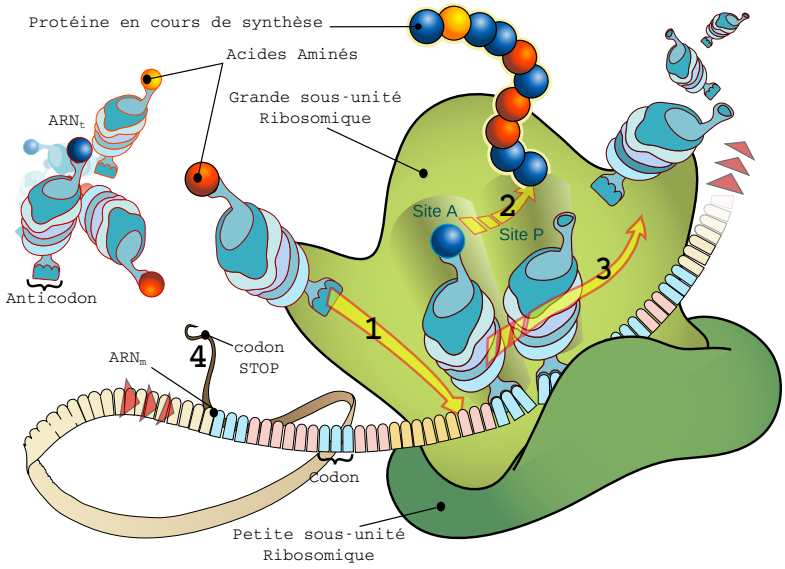


Figure 1.12 – *Traduction de l'ARNm en protéine*[13]. *Reconnaissance par l'anticodon de l'ARNt du codon en cours de lecture par le ribosome (1), élongation de la synthèse protéique (2), décalage de la phase de lecture du ribosome au codon suivant (3) et terminaison de la synthèse protéique à la rencontre du codon STOP (4).*

L'initiation Même si quelques divergences existent entre l'initiation chez les eucaryotes et les procaryotes, les mécanismes principaux demeurent communs. Ainsi la petite sous-unité ribosomique permet l'interaction entre le codon initiateur de la traduction de l'ARNm et l'ARNt initiateur portant (1) l'anticodon complémentaire au codon START AUG et (2) le premier acide aminé qui compose la molécule et correspond au codon START, la méthionine.

Chez les eucaryotes la coiffe à l'extrémité 5′ des ARNm permet la reconnaissance par un complexe d'initiation de la traduction qui recrute la petite sous-unité ribosomique et de l'ARNt de démarrage. Ce complexe nouvellement formé glisse alors vers l'extremité 3′, vers le premier codon de l'ARNm. Le déclenchement de la traduction a lieu et la seconde sous-unité ribosomique est recrutée.

12. D'après une illustration de Mariana Ruiz Villarreal.

L'élongation Le ribosome se déplace le long de l'ARNm en associant chaque codon lu à l'anti-codon d'un ARNt. Ce processus permet ainsi l'élongation du peptide en cours de synthèse par agrégation d'acides aminés. L'ARNt ayant libéré son acide aminé est alors détaché du complexe dans le cytoplasme pour laisser la place à l'ARNt complémentaire au codon suivant. Ainsi de suite la chaine polypeptidique s'allonge suivant l'ordre donné par les codons de l'ARNm. Du point de vue énergétique, la formation de la liaison peptidique entre chaque acide aminé ne nécessite aucune énergie extérieure au système supplémentaire à celle contenue dans la liaison précédente de l'acide aminé à l'ARNt.

La terminaison Quand le ribosome parvient au niveau d'un codon STOP ne codant pour aucun acide aminé, des facteurs de terminaison vont intervenir pour effectuer le relargage de l'ensemble des facteurs et le recyclage du ribosome via la dissociation de ses sous-unités. Si l'action de ces facteurs a pu être observée par différentes techniques leur fonctionnement exact reste encore largement méconnu. La séquence polypeptidique alors synthétisée peut être soumise à d'ultimes modifications ou être fonctionnelle directement.

La chaine polypeptidique La chaîne polypeptidique synthétisée au cours du processus de traduction est une macromolécule biologique d'acides aminés. Les acides aminés, molécules de base des protéines, sont des composés chimiques possédant deux groupes fonctionnels : un groupe amine $-NH_2$ et un groupe carboxyle $-COOH$. Les acides aminés successifs d'une protéine sont reliés par une liaison peptidique, d'où le nom de chaîne peptidique, issues de la réaction de condensation des deux groupes fonctionnels. Tout comme les molécules d'ADN et d'ARN, les chaînes peptidiques sont orientées selon leurs groupes fonctionnels, de l'extrémité $-NH_2$ vers l'extrémité $-COOH$.

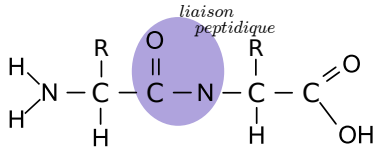


FIGURE *1.13 – Liaison peptidique entre deux acides aminés provenant d'une réaction de condensation des extrémités $-COOH$ et $-NH_2$.*

En général, une protéine contient au moins 40 acides aminés, un peptide est une chaîne de plus petite taille. En outre une protéine (ou un peptide) peut être composée d'une ou plusieurs chaînes. La ou les chaîne(s) polypeptidique(s) qui constitue(nt) une protéine adopte(nt) une conformation spatiale particulière selon leur composition et leur environnement. On distingue trois niveaux organisationnels communs à toutes les chaînes polypeptidiques :

-Structure primaire : Elle est donnée par l'ordre dans lequel les acides aminés s'enchaînent (celui de la séquence primaire), codé dans le génome.

-Structure secondaire : Elle est constituée par le repliement de la séquence primaire sur elle-même qui permet la formation de structures secondaires parmi les hélices, feuillets, boucles, coudes...Ces structures sont formées suite à la

création de liaisons hydrogènes entre les atomes de carbone et d'azote de deux liaisons peptidiques voisines.

-Structure tertiaire : Elle provient de l'agencement des structures secondaires entre elles. Cette structure est souvent renforcée par l'établissement de ponts disulfures, par exemple.

Un quatrième niveau peut être défini lorsqu'une protéine est composée de 2 à n chaines polypeptidiques :

-Structure quaternaire : Elle décrit l'agencement relatif des sous-unités formées par chaque chaîne polypeptidique les unes par rapport aux autres. En fonction du type des chaînes assemblées, on parle d'*homodimère* si toutes les chaînes polypeptidiques sont identiques ou d'*hétérodimères* si au moins deux chaînes sont différentes.

Cette structure complexe liée à l'agencement dans l'espace des acides aminés influe sur la fonction de la protéine.

Les protéines interviennent à plusieurs niveaux dans une cellule et assurent la majorité des fonctions cellulaires. Elles ont un rôle structurel, un rôle dans la mobilité, un rôle catalytique, un rôle dans la régulation et l'expression des gènes...

Les modifications post traductionnelles Après sa synthèse, une protéine peut subir des modifications chimiques, le plus souvent réalisées par des enzymes, qui induisent un changement de la fonction de cette protéine.

On recense quatre types de modifications post-traductionnelles :

- Ajout d'un groupe fonctionnel (acétylation, glycosylation...)
- Addition d'un groupe peptidique (ubiquitination...)
- Modification de la nature chimique d'un acide aminé (arginine en citrulline...)
- Modification structurale (clivage...)

1.1.3 Le Gène Unité de Base des Fonctions Cellulaires

L'ensemble de la molécule d'ADN n'est pas transcrite au cours de la transcription. Seules certaines régions de la molécule d'ADN, délimitées par des séquences particulières, seront transcrites en ARN. Ces séquences sont appelées gènes. L'ensemble des séquences géniques et des séquences entre les gènes (séquences intergéniques) d'un individu donné constitue ce qu'on appelle son *génome*[14].

La séquence d'ADN présente une importance dans la compréhension de l'activité cellulaire. Le *séquençage* permet de connaître l'enchaînement des nucléotides et donc de cartographier cette molécule d'ADN. C'est dans cette optique que des projets tels que le *Projet Génome Humain* (*HGP*) ont vu le jour (Collins and al., 2004).

14. On note que certains virus, les virus à ARN, ne possèdent pas de molécule d'ADN et ne présentent pas d'étape de transcription. Par extension on appelle gène les fragments d'ARN traduits en protéines et l'ensemble de ces séquences constituent leur génome.

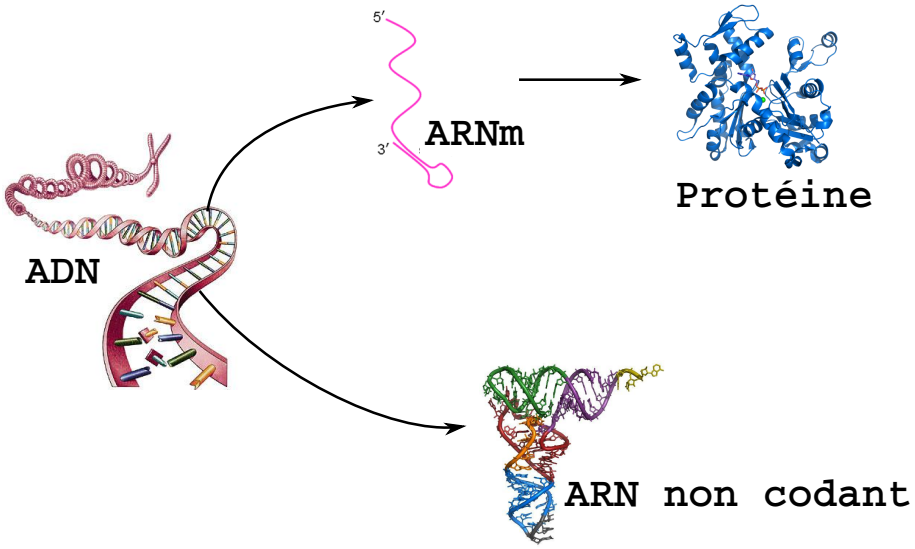


FIGURE 1.14 – *De la séquence ADN à la séquence ARN et peptidique.*

C'est en 1985, suite au séquençage du premier génome d'un organisme biologique, le virus bactériophage $\phi X174$ (Sanger et al., 1977a) en 1977, que Renato Dulbecco propose l'idée du séquençage du génome humain. Mais ce n'est que 5 ans plus tard, en 1989, que ce projet public international verra vraiment le jour sous la direction de James Watson (basé sur la méthode de contiguage). A cet instant le nombre de gènes dits codant présents sur cette séquence ADN d'environ 3 milliards de bases est évalué entre 30 et 100000. En parallèle, un autre projet concurrent privé est lancé : le projet Venter (basé sur la méthode shotgun).

Au cours de ces projets de nombreuses innovations scientifiques voient le jour et des méthodes de séquençage plus rapides et plus efficaces sont mises en place à partir de l'automatisation de Sanger (Sanger et al., 1977b). Une première version brute est déposée dans les bases publiques en février 2001 et le génome complet est mis à disposition de la communauté scientifique en 2004. On dénombre alors environ 20000 gènes pour $3,2$ milliards de bases constituant une molécule d'ADN de 2 mètres.

En 2005, une nouvelle révolution technologique a eu lieu avec les *NGS* (« New Generation Sequencing ») en réduisant les coûts et le temps de séquençage et augmentant par là-même la quantité de données à analyser.

(i) Séquences Géniques

Au sein des séquences géniques on distingue deux types de séquences : les séquences *codantes* et les séquences *non codantes*(voir Figure 1.14). Ces deux types de séquences sont identifiables au sein de la molécule d'ADN.

Gènes codants Un gène commence par une séquence nucléotidique appelée *promoteur* et se termine par une séquence terminatrice appelée *terminateur* (voir la Figure 1.15). Alors que le promoteur a pour rôle de permettre l'initiation de la transcription, le terminateur permet au contraire la terminaison de la transcription. Le promoteur présente également un rôle de régulateur de la transcription et permet de moduler l'expression des gènes. Entre deux séquences de terminaison, soit entre deux codons stop on définit une *ORF* (*Open Reading Frame*).

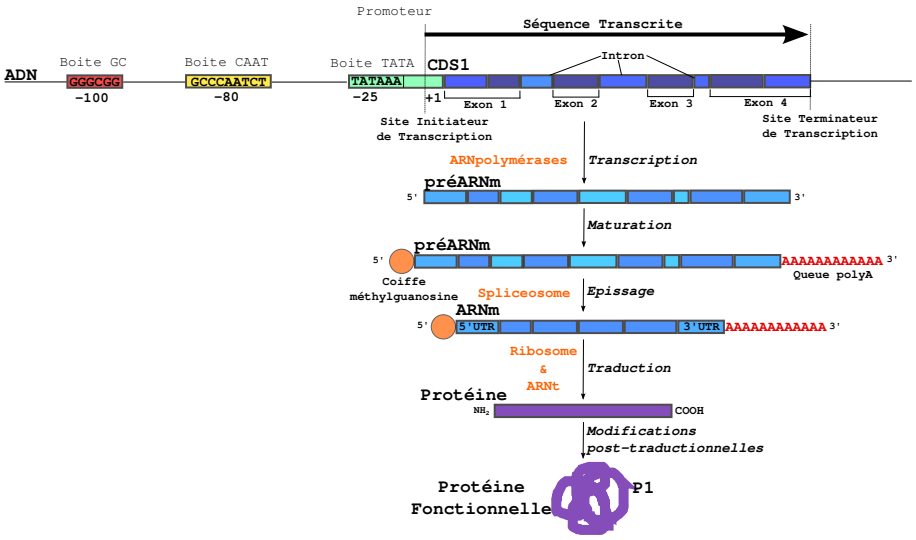


Figure 1.15 – *Architecture morcelée en exons et introns des gènes eucaryotes.*

De plus, la région transcrite d'un gène présente au début un codon start [15], souvent précédé d'une région 5′utr, et à la fin un codon stop [16]. On appelle *CDS* (Coding DNA Sequence, voir Figure 1.15) la séquence comprise entre ces deux codons. Ces CDS ont la particularité d'être en premier lieu *transcrites* en une nouvelle molécule, l'Acide RiboNucléique messager ou *ARNm*, qui est ensuite elle-même *traduite* en protéine : on parle, respectivement, de la *transcription* (voir la section (i)) et de la *traduction* (voir la section (ii)) des gènes. Si la CDS est seulement transcrite en ARN on parle de séquence non codante et d'ARN non codant (ARNnc). La fonction de ce gène est alors réalisée par son ARNnc.

Suivant le type cellulaire, on distingue deux types de gènes. Les gènes eucaryotes présentent une organisation morcelée et sont transcrits pour la majorité d'entre eux gène à gène. L'ARNm transcrit contient donc une seule et unique CDS. La CDS de ces gènes se compose d'une alternance de séquences *exoniques* et de séquences *introniques*. Alors que les séquences exoniques pourront être retrouvées dans la séquence protéique codée par la CDS, les introns seront eux excisés au cours d'une étape d'*épissage*. Contrairement aux eucaryotes, les gènes procaryotes présentent souvent

15. Le triplet aug est le codon aug universel mais des codons stars alternatifs existent.
16. on dénombre trois codons universels : uag, uaa et uga

une organisation continue. La CDS est retrouvée entièrement dans la séquence protéique. Les gènes procaryotes sont organisés en *opérons*. Un opéron regroupe entre l'initiateur et le terminateur de transcription plusieurs gènes adjacents. Dans ce cas, l'ARNm transcrit contient plusieurs CDS, on parle d'ARNm polycistronique.

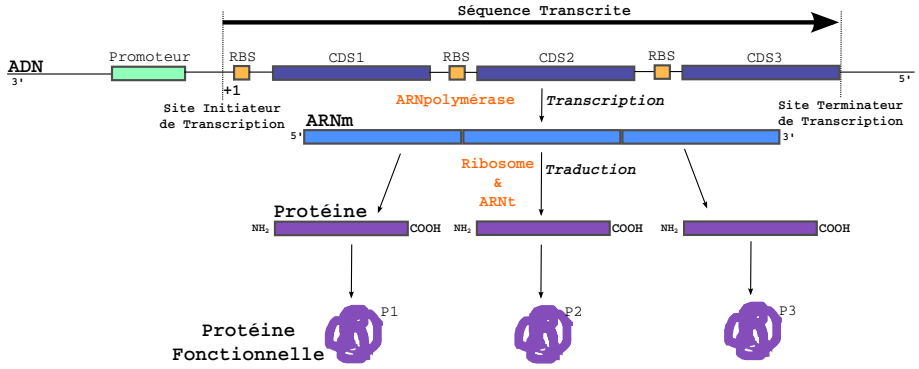


FIGURE *1.16 – Architecture continue de trois gènes procaryotes au sein d'une structure en opéron.*

L'ADN, par l'action de ses gènes codants et des protéines qu'ils codent, contrôle les mécanismes du métabolisme via les protéines qui s'y trouvent. Mais l'ensemble des gènes n'est pas exprimé de manière simultanée et continue. En outre, tous les gènes ne sont pas exprimés dans chaque cellule d'un organisme pluricellulaire. C'est ainsi que s'opère la spécialisation des cellules : par l'expression différentielle des gènes qu'elle possède. Cette expression différentielle est, entre autres, permise par une régulation via certaines séquences non codantes.

Gènes non codants Alors que les séquences des gènes codants sont transcrites en ARNm traduits en protéines fonctionnelles, les séquences des gènes non codants sont transcrites en ARN mais ceux-ci ne sont pas traduits en protéines.

Ces séquences non traduites en protéines ont un rôle fonctionnel via leur forme intermédiaire : l'Acide RiboNucléique. On parle alors d'ARN non codant ou *ARNnc*. L'ARN est une molécule chimiquement proche de l'ADN et est synthétisé à partir d'une séquence ADN dont il sera une copie complémentaire. Contrairement à l'ADN confiné dans le noyau cellulaire, l'ARN, plus petit, a la possibilité de sortir par les pores nucléaires et de rejoindre le cytoplasme cellulaire.

L'importance de ces ARNnc a longtemps été sous estimée tant quant à leur rôle dans l'activité cellulaire que quant à la proportion du génome qu'ils occupent. En effet, chez l'Homme seul 1,2% du génome est traduit en protéines.

Au sein des ARNnc il existe une grande diversité (Buckingham et al., 2003; Mallick and Ghosh, 2012) (voir Figure 1.17). Chacune des différentes classes d'ARNnc présente ses propres propriétés et ses propres fonctions.

17. aucune distinction entre les différents embranchements phylogénétiques n'est réalisée ici, par exemple les ARNxi n'ont à ce jour été observés que chez les mammifères

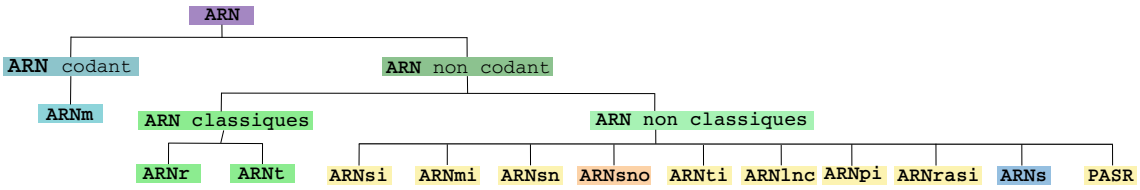


Figure 1.17 – *Représentation graphique non exhaustive des différentes sous familles d'ARN. On distingue alors les ARNnc appartenant aux eucaryotes (jaune), aux procaryotes (bleu) et aux archae (rouge)*[17].

ARNnc	Organisme	Taille	Fonction
ARNsi	eucaryotes	$21-22$	Régulation post transcriptionnelle Contrôle des transposons
ARNmi	eucaryotes	$18-25$	Régulation post transcriptionnelle des gènes
ARNti	eucaryotes	$18-22$	Régulation des modifications de la chromatine Régulation des protéines impliquées dans l'initiation de la transcription
ARNpi	eucaryotes	$24-30$	Régulation des transposons Régulation de la méthylation de l'ADN Régulation de l'état de la chromatine
ARNsno	eucaryotes, archae	$80-200$	Impliqué dans les modifications chimiques (méthylation, pseudouridylation) des ARNr et ARNt
ARNpsno	eucaryotes	$20-100$	Régulation de l'épissage des préARNm (régulation de l'expression génique)
ARNs	bactéries	$50-300$	Régulation post transcriptionnelle des gènes
moRNA	eucaryotes	~ 20	Fonction inconnue
RNAtels	eucaryotes	~ 24	Implication dans le maintien des télomères
ARNnatsi	eucaryotes	$21-24$	Régulation des gènes de réponse au stress
ARNcrasi	eucaryotes	$34-42$	Impliqués dans les modifications de l'état chromatinien Impliqués dans la formation et la maintenance des centromères
ARNrasi	eucaryotes	$24-29$	Maintien de la structure hétérochromatinienne
ARNhcsi	eucaryotes	24	Impliqués dans la méthylation des histones et de l'ADN
ARNscn	eucaryotes	28	Impliqué dans les modifications de la chromatine
ARNqi	eucaryotes	$20-21$	Impliqué dans l'inhibition de la synthèse de protéines spécifiques
ARNcasi	eucaryotes	24	Implication dans la méthylation de la cytosine Implication dans les événements de transcription
ARNtasi	eucaryotes	21	Régulation post transcriptionnelle
ARNlnc	eucaryotes	> 200	Régulation de phénomènes épigénétiques
ARNxi	eucaryotes	$25-42$	Inactivation du chromosome X
PASR	eucaryotes	$22-200$	Régulation de l'expression des gènes
TASR	eucaryotes	$22-200$	Fonction inconnue
non PASR	eucaryotes	–	Fonction inconnue
TASRa	eucaryotes	< 200	Fonction inconnue

Table 1.1 – *Aperçu de diverses familles ARN extrait de Mallick and Ghosh (2012)*

ARN ribosomique (ARNr) L'ARNr (dont la taille varie d'une centaine de nucléotides à plusieurs milliers de nucléotides) est le principal constituant du complexe ribonucléoprotéique, le ribosome, acteur de la traduction (voir Figure 1.12).

ARN de transfert (ARNt) Les ARNt sont de petits ARN de 70 à 100 nucléotides. Ils interviennent comme intermédiaires au cours de la traduction des ARNm en protéines, en permettant la lecture du code génétique. Chaque ARNt est spécifique d'un acide aminé et de l'anticodon qui lui correspond (voir 1.12). Ils interviennent également au niveau de la réplication de l'ADN, l'épissage et la régulation de la traduction (Giegé et al., 1998).

Autres *ARNnc* Comme il est possible de l'observer dans la Table 1.1, le grand nombre d'ARNnc découverts fait l'objet de nombreuses études portant sur la compréhension de l'implication de ces séquences dans les mécanismes cellulaires. En effet, s'il est connu que les ARN ont un rôle dans l'activité cellulaire, leurs mécanismes d'action le sont moins.

(ii) Séquences Intergéniques

D'autre part entre chaque région codante est définie une séquence intergénique ou intergène. Ces intergènes ont longtemps été considérés comme des portions d'ADN *inutiles*. Mais en réalité ils peuvent contenir des séquences impliquées, entre autres, dans la régulation de l'expression des gènes à proximité.

Ces séquences non traduites de l'ADN peuvent également contenir des *reliques* d'anciens gènes codants. Certaines d'entre elles ont perdu toute implication dans l'expression génique mais d'autres, nommées *pseudogènes*, non. Mis en évidence au sein des séquences non codantes de l'ADN dans les années 1970 (Jacq and Brownlee, 1977), les pseudogènes ont longtemps été considérés comme des séquences sans impact dans l'activité cellulaire. Néanmoins il a été mis en évidence que les pseudogènes avaient pour certains un rôle dans la régulation de gènes via leur ARNnc (Kamalika and Tapash, 2013).

Les pseudogènes étant le point d'ancrage d'une partie de ce manuscrit, une étude plus approfondie de ces séquences est proposée à la section 1.3.2.

Les séquences intergéniques, pseudogènes exclus, demeurent encore à ce jour peu connues.

(iii) Mécanismes de Régulation de l'Expression des Gènes

La cellule a développé des mécanismes qui lui permettent de réprimer ou d'activer l'expression des gènes selon les conditions environnementales de la cellule (stress ou autre). Cette régulation permet, entre autres, à l'organisme d'adapter son métabolisme à son environnement mais surtout, il permet l'expression différentielle du génome selon la spécialisation de la cellule ou la période du développement cellulaire.

Au cours de l'expression des gènes toutes les étapes en partant de la séquence ADN jusqu'au produit final, protéine ou ARN, sont régulées par divers mécanismes. Ainsi la transcription puis la traduction mais aussi les étapes de maturation et les produits eux-même sont soumis à des mécanismes de régulation permettant de moduler, d'accroître ou de décroître, la quantité d'ARN et de protéines synthétisés.

Tout d'abord, il existe différents niveaux de compaction de l'ADN. En particulier, l'ADN peut être présent sous une forme lâche (appelée *euchromatine* chez les eucaryotes) ou sous une forme condensée (appelée *hétérochromatine* chez les eucaryotes). Alors que l'ADN lâche est facilement accessible par des protéines, l'ADN condensé n'est pas accessible aux polymérases [18].

La transcription dépendant de facteurs de transcription leur présence ou absence influe sur le taux de transcription. On en distingue deux principales classes (Williams et al., 2010) :

- Les éléments cis-régulateurs sont des séquences ADN de 6 à 15 nucléotides de long le plus souvent en amont de la séquence codante à environ 5 000 nucléotides du gène d'intérêt en moyenne.
- Les éléments trans-régulateurs sont des facteurs de transcription se fixant spécifiquement aux régions cis-régulatrices de manière à activer ou inhiber la séquence codante. Elles sont généralement situées sur un autre chromosome.

Le dernier niveau de régulation, la régulation post-traductionnelle, réfère au contrôle de la quantité de protéines actives, par la régulation de l'expression du gène codant cette protéine ou de sa stabilité. La protéine produite peut ainsi être modifiée chimiquement. Ces modifications peuvent jouer sur sa conformation spatiale et donc sur son activité. Enfin des complexes enzymatiques peuvent détruire ces protéines.

On différencie ensuite certaines particularités de la régulation de l'expression (transcription et traduction) des gènes chez les procaryotes et chez les eucaryotes par des mécanismes différents.

Régulation procaryote Certains gènes procaryotes peuvent être regroupés au sein de structures opéroniques. Un opéron (voir Figure 1.2) est une unité d'expression et de régulation des gènes bactériens. Un opéron permet le regroupement dans l'espace, sur le même chromosome, de plusieurs gènes. Chaque opéron est précédé d'un opérateur, c'est-à-dire d'une séquence composée d'un promoteur et d'un *RBS* (*Ribosome Binding Site*), permettant la régulation de l'opéron, soit la régulation simultanée de toutes les CDS qui composent l'opéron.

Chez les procaryotes, la régulation post traductionnelle est peu présente. Elle consiste principalement en une rétro régulation d'une protéine sur sa propre traduction.

18. On note qu'en changeant les zones qui sont condensées, la cellule contrôle quels gènes sont exprimés.

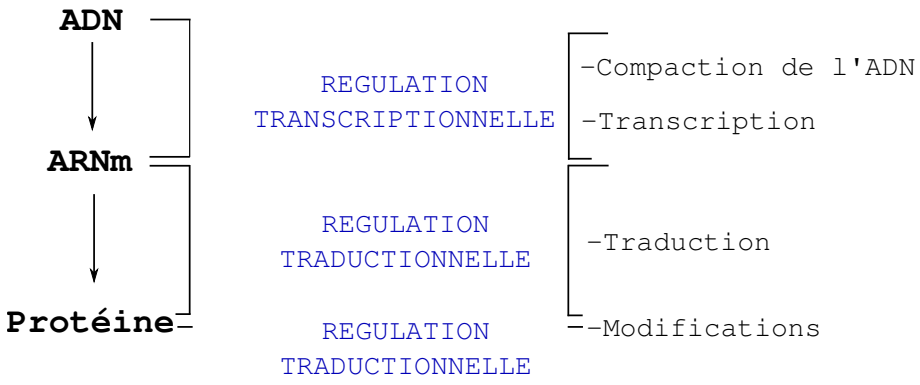


FIGURE 1.18 – *Les différents points de régulation de l'expression génique chez les procaryotes.*

Régulation eucaryote Chez un organisme multicellulaire, l'expression variable des gènes est à l'origine de la spécialisation cellulaire. On compte par exemple 250 types cellulaires différents marqués par leur morphologie, leur biochimie, leur rôle dans l'organisme...

On distingue cinq niveaux de régulations listés dans la Figure 1.19.

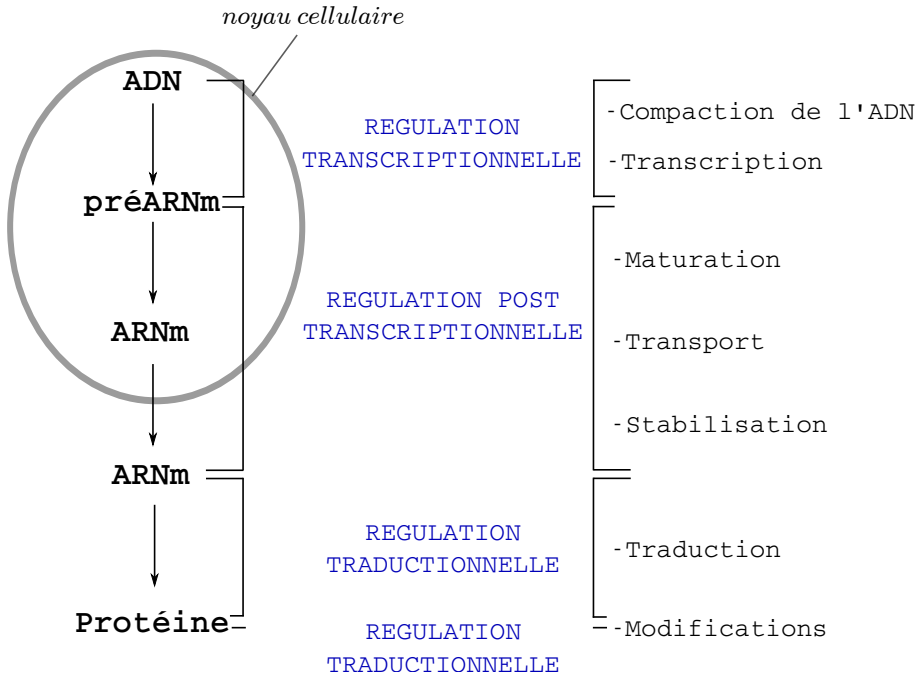


FIGURE 1.19 – *Les différents points de régulation de l'expression génique chez les eucaryotes.*

Le pré-ARNm produit par la transcription n'est pas encore mature et présente une durée de vie relativement courte du fait de la présence de protéines de dégradation des ARN. Les modifications post-transcriptionnelles ont pour objectif de stabiliser la molécule. Toute altération de ces processus aura un impact sur la traduction

de l'ARN et donc sur l'expression du gène.

Le phénomène d'épissage permet d'exciser les introns de la molécule de pré-ARNm. Cependant, il existe un épissage alternatif qui en outre excise certains exons, ce qui induit une traduction en une protéine différente.

Les mécanismes de régulation de la traduction agissent majoritairement au niveau du démarrage du décodage de l'ARNm par le ribosome. La traduction peut être régulée par des protéines se fixant sur la séquence initiatrice de la traduction de l'ARNm bloquant alors la liaison du ribosome et donc la traduction de cet ARNm. Elle peut également être modulée par des ARN anti-sens qui s'apparient à l'ARNm pour former un double brin ne pouvant pas être traduit. Enfin selon la composition du cytoplasme le repliement de l'ARNm peut être modifié et la structure secondaire nuire à la traduction.

1.2 L'Évolution des Génomes

La fonction d'un ARN ou d'une protéine est portée par sa séquence de bases provenant du gène dont il est issu. Toutefois pour un ARN ou une protéine donnés, entre deux espèces, la séquence génique correspondante n'est pas nécessairement identique. Ainsi, au cours des générations les séquences géniques évoluent sous l'effet de facteurs extérieurs. C'est le moteur de la diversité au sein d'une population et de l'évolution des espèces.

Dans une population d'individus d'une même espèce il existe des différences plus ou moins importantes entre ses individus. Toute différence visible est appelée caractère et plusieurs traits existent pour un même caractère. Chez un individu donné ses traits de caractères constituent son phénotype. Ce phénotype est directement lié au génotype de l'individu. Les différences phénotypiques traduisent alors des différences génotypiques [19].

1.2.1 Mécanismes de Divergence des Génomes

Les diversités phénotypiques proviennent de la divergence des génomes suite à des événements évolutifs non conservatifs. Des phénomènes évolutifs conservatifs qui n'impactent ni le génotype ni le phénotype peuvent également avoir lieu.

Nous nous focaliserons ici sur les événement ayant des conséquences sur le génotype.

19. On remarque que des phénotypiques similaires ne traduisent pas nécessairement des génotypes similaires

(i) Événements de Modification

Mutations ponctuelles Toute modification affectant au moins un nucléotide (mais jusqu'à une dizaine environ) est appelée mutation ponctuelle. Au sein d'une population, si ces variations nucléotidiques interviennent dans un gène, alors on parle de *polymorphisme*. On appelle *SNP* (« Single Nucleotide Polymorphisme ») toute modification d'un nucléotide dans la séquence ADN. De plus si ces *substitutions* induisent une modification du phénotype, on parle d'allèle de gène. On note que s'il existe au moins deux allèles pour un gène, ce gène est dit polymorphe.

Pour être exprimée, c'est-à-dire pour avoir un impact sur le phénotype, une mutation doit être une substitution modifiant le codon initial.

On recense trois types de substitution :

- Silencieuse : de part la redondance du code génétique, la substitution n'affecte pas la traduction du codon.
- Faux-sens : la substitution modifie le codon par un autre codon. Si elle touche une séquence codante, la protéine peut être plus ou moins modifiée selon la position de la mutation.
- Non-sens : la substitution modifie le codon par un codon STOP. Si la mutation touche une séquence codante, la protéine produite est alors tronquée.

On observe alors que les mutations silencieuses n'ont aucune conséquence phénotypique contrairement aux deux autres substitutions non silencieuses qui peuvent se traduire par des altérations du phénotype.

Réarrangements Les remaniements chromosomiques concernent le nombre ou la structure des chromosomes et peuvent impliquer un ou plusieurs chromosomes. On distingue différents types de remaniements équilibrés, c'est-à-dire sans perte ni gain de matériel génétique.

Les recombinaisons intra-chromosomiques, ou enjambement, sont l'échange de segments entre deux chromosomes homologues au niveau de sites précis appelés chiasmas chez les eucaryotes. Pour chaque paire de chromosomes on dénombre entre 1 et 5 chiasmas. Selon la localisation et la longueur du fragment impliqué, c'est-à-dire inséré ou supprimé, les conséquences sont variables.

La translocation est une mutation caractérisée par l'échange de fragments chromosomiques entre deux chromosomes non homologues.

L'inversion résulte de la cassure d'un fragment d'un chromosome suivie d'une rotation de 180° de ce fragment et de sa réinsertion dans le même chromosome. Ce type de réarrangement est souvent invisible au niveau phénotypique.

(ii) Événements de Perte ou de Gain

Redondances Il existe plusieurs mécanismes permettant la duplication d'un ou plusieurs gènes, d'un chromosome voire même du génome entier (phénomène rare).

Duplication La duplication est une mutation génétique caractérisée par le doublement du matériel génétique. Elle peut concerner une large séquence chromosomique, un gène ou bien une suite de quelques nucléotides. Lorsqu'elle concerne un gène, cette duplication crée une copie supplémentaire affranchie de la pression de sélection et laisse la possibilité à la copie de muter sans conséquences nuisibles à l'organisme. Cependant, à cause de cet excès de données génétiques, elles peuvent aussi conduire à des problèmes au cours du développement ou contribuer à la croissance de tumeurs (Wong et al., 1986).

Les duplications de gènes sont des événements fréquents. Ainsi, au sein d'une même espèce, pour deux individus donnés, le nombre de gènes peut varier. On parle pour ce phénomène de polymorphisme du nombre de répétitions.

Polyploïdie La polyploïdie résulte de la duplication complète d'un génome. Cet événement est plutôt rare comparé au phénomène de duplication de gènes. On peut citer par exemple le blé qui présente 6 copies de son génome, il est hexaploïde.

Transpositions Au sein du génome des éléments *transposables* sont réinsérés dans le génome par rétrotranscription de leur ARN. Cependant au cours de leur transcription une portion de la séquence adjacente, comprenant éventuellement un ou plusieurs gènes, peut également être transcrite. Ainsi, lors de la rétrotranscription au sein du génome de cet ARN, les gènes transcrits par accident sont réinsérés dans le génome.

Insertion et Délétion Les phénomènes d'insertion et de délétion peuvent être observés à différentes échelles du génome.

Au niveau des nucléotides l'insertion comme la délétion d'un nucléotide entraîne un décalage de la phase de lecture. Si la séquence est codante, cela peut la modifier de deux manières possibles : (1) cela peut générer l'apparition d'un codon stop et le médiateur cellulaire sera donc tronqué, (2) cela peut décaler la lecture des codons et modifier entièrement la fin du médiateur cellulaire. Dans les deux cas ces insertions ou délétions peuvent avoir des conséquences phénotypiques.

La délétion peut également toucher les chromosomes et peut être de taille variable. Les conséquences d'un tel remaniement dépendent alors de la longueur et des gènes amputés.

Transferts horizontaux Le transfert horizontal est un mécanisme d'échange de matériel génétique entre deux organismes. Lorsqu'un organisme transducteur (virus, parasite, mitochondrie...) transfère une séquence contenant un certain nombre de séquences codantes dans un nouveau génome, cette séquence porte le nom d'*îlot génomique* (Hacker and Carniel, 2001). Ainsi, chez les procaryotes, un îlot génomique est une séquence présente dans le génome de certaines souches d'une espèce et absente des autres souches de la même espèce ou d'espèces proches (ce qui implique un transfert horizontal récent).

Fusion et fission L'ensemble des phénomènes de duplication, de réarrangement, de transposition, d'inversion, de transfert horizontaux, ...peuvent s'accompagner de deux phénomènes parallèles (Durrens et al., 2008) :

- la fusion des séquences de part et d'autre de la région excisée ou la fusion des séquences aux extrémités de la région insérée avec les séquences adjacentes.
- la fission de la séquence au milieu de laquelle est insérée la région déplacée.

Selon les séquences impliquées dans les phénomènes de fusion et de fission l'impact sur le phénotype peut varier.

Ces deux phénomènes rendent particulièrement compte des notions de naissance de gènes (pour la fusion) et de sénescence de gènes (pour la fission).

1.2.2 Dynamiques d'Évolution des Génomes

La conservation de l'intégrité de l'information génétique est nécessaire à la survie de l'individu. Pour cela, de nombreux mécanismes cellulaires de réparation de l'ADN assurent cette conservation. Toutefois, ces mécanismes peuvent défaillir et engendrer les mécanismes décrits ci-avant. S'ils peuvent porter atteinte à l'intégrité cellulaire, ces mécanismes peuvent, à plus long terme, être indispensables à la survie d'une population en générant une variabilité génétique nécessaire à l'adaptation à un environnement changeant.

(i) Plasticité

Une propriété importante de la molécule d'ADN est cette capacité à subir des modifications à diverses échelles du génome qui peuvent diversifier la combinaison des gènes ou leur niveau d'expression. On parle de plasticité des génomes. La comparaison de génomes appartenant à une même espèce ou à un même genre (ou quelque soit l'échelle de la classification choisie) permet de mettre en évidence des régions similaires et des régions dissimilaires.

Familles multigéniques Une famille multigénique désigne un ensemble de gènes qui présentent des homologies de séquences et sont issus d'un même gène ancestral. Ainsi les protéines issues de ces gènes auront globalement les mêmes fonctions (mais avec le temps et l'accumulation des mutations celles-ci peuvent finir par diverger) (Wajcman et al., 2009). De tels gènes sont appelés gènes homologues. On distingue alors les paralogues des orthologues.

Définition *Homologues : Paralogues/Orthologues* *Soit deux séquences S_1 et S_2, ces deux séquences sont dites homologues si elles dérivent d'une séquence ancestrale commune S_A.*

- *Si S_1 et S_2 résultent d'une duplication de S_A (S_1 et S_2 peuvent donc appartenir à la même espèce) : on parle de **séquences paralogues**.*
- *Si S_1 et S_2 appartiennent à deux espèces différentes et ont évolué à partir d'une séquence unique appartenant au dernier ancêtre commun des deux espèces : on parle de **séquences orthologues**.*

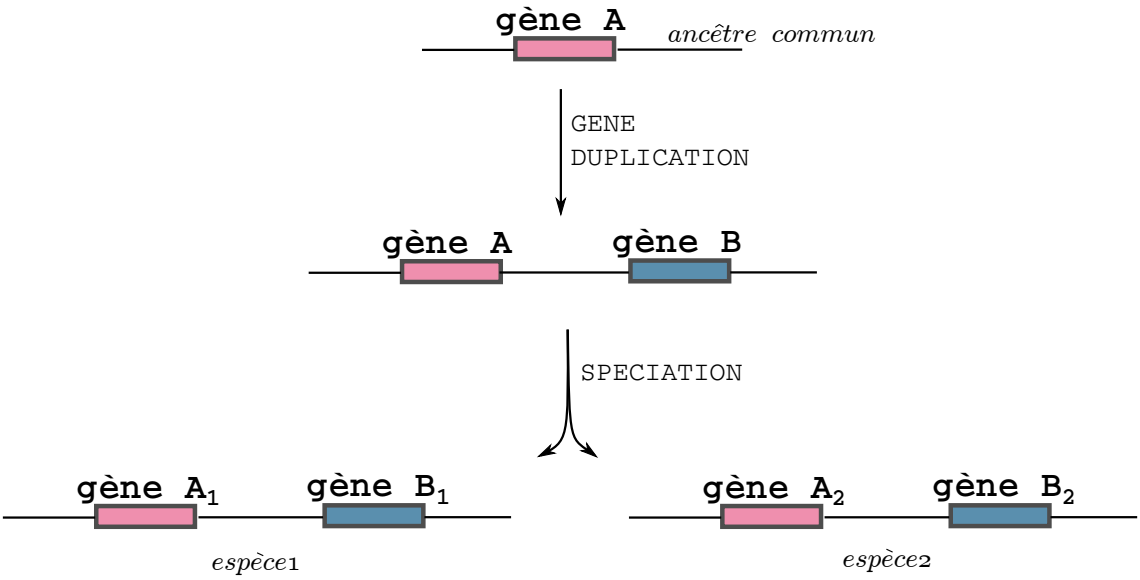


Figure 1.20 – *Séquences homologues. Les gènes A et B sont issus de la duplication d'un même gène ancestrale : ils sont* paralogues. *A_1, A_2 et B_1, B_2 proviennent d'un événement de spéciation : ils sont orthologues.*

Gènes paralogues Deux gènes paralogues sont deux gènes homologues, issus de la duplication d'un gène ancestral, qui ont divergé au sein d'une même espèce. L'une, au moins, des copies a alors développé de nouvelles fonctions (voir Figure 1.20).

Gènes orthologues Deux gènes orthologues sont deux gènes homologues issus d'un même gène ancestral présent chez le dernier ancêtre commun aux deux espèces auxquelles appartiennent chaque gène. Suite à un événement de spéciation chacun des gènes a alors évolué séparément. Il n'est pas impossible que les deux gènes orthologues aient une fonction différente.

Tout comme deux gènes paralogues ne sont pas nécessairement situés sur le même chromosome, deux gènes orthologues peuvent être situés à deux emplacements différents dans le génome (*loci*). Ce phénomène de déplacement des séquences au sein du génome est appelé *translocation*.

Pan génome La plasticité génique peut être étudiée à plusieurs échelle : du point de vue intra-spécifique, soit entre les diverses souches d'une même espèces, ou du point de vue inter-spécifique, soit entre différentes espèces.

L'étude des familles multigéniques et de la composition des génomes permet d'identifier des séquences conservées, le *core génome*, et des séquences propres à chacun des génomes, le *génome accessoire*. Suivant l'échelle choisie, les core génomes et génomes accessoires varient. En effet, plus l'échelle est élevée plus les génomes ont divergé. Le *pan génome* est alors la somme du core génome et de la somme de tous les génomes accessoires.

(ii) Adaptation

La plasticité du génome est une source de diversité pour l'adaptation des génomes à un nouvel environnement. Ainsi, les traits héréditaires permettant une meilleure adaptation à l'environnement et donc favorisant la survie et la reproduction des individus les possédant présentent une fréquence d'apparition croissante dans la population au fil des générations. Cette sélection naturelle à laquelle est soumis tout individu est couplée à la pression de sélection, c'est-à-dire à un ensemble de contraintes environnementales auxquelles est assujettie une population : les facteurs physico-chimiques ou biotopes (climat, milieu...) et les autres êtres vivants ou biocénose (prédateurs, parasites...).

Le lien entre génotype et phénotype est particulièrement présent ici puisque sans les modifications du génotype aucune modification du phénotype n'aurait pu être observée. Ces traits s'héritent et se transforment, on parle de *dérive génétique* et ils s'acquièrent aussi (par exemple par transfert horizontal). L'ensemble de ces mécanismes à l'origine de l'évolution peut alors dans certains cas être à l'origine de l'apparition de nouvelles espèces (on parle de *spéciation*).

1.3 Les Objets Biologiques Non Codants

Outre les gènes sur lesquels les recherches scientifiques se sont longtemps concentrées, l'ADN présente d'autres éléments qui jouent un rôle essentiel dans la régulation de l'activité des gènes mais aussi dans l'apparition de certaines pathologies (Yan and Wang, 2012). Nos travaux se focalisent sur les deux objets non codants que sont les ARN et les pseudogènes que nous décrivons plus en détails dans cette section.

1.3.1 Les Acides RiboNucléiques ou ARN

(i) Definition

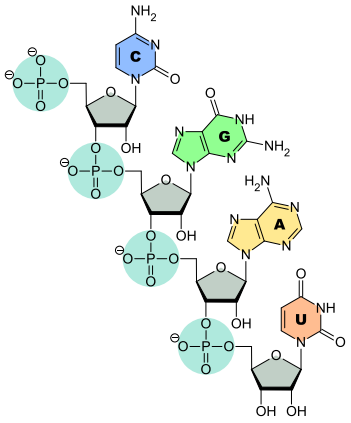


Figure 1.21 – *Fragment d'une molécule d'ARN composée des quatre nucléotides A, C, G et U.*

Comme il a été vu dans le paragraphe (i), un ARN est un polymère linéaire constitué de l'enchaînement de nucléotides (voir Figure 1.21). Un nucléotide comprend un pentose, le ribose, une base azotée, parmi l'adénine (A), la guanine(G), l'uracile(U) ou la cytosine(C), et un groupement phosphate permettant la liaison, une liaison phosphodiester, des nucléotides entre eux.

De nombreuses similarités peuvent être relevées entre ADN et ARN, cependant des différences importantes sont à noter :

(1) l'ARN contient un ribose et non un désoxyribose, le rendant plus instable.

(2) l'uracile possédant les mêmes propriétés d'appariement à l'adénine remplace la thymine de l'ADN.

(3) l'ARN est majoritairement rencontré sous sa forme simple brin [20].

(4) les molécules d'ARN sont plus courtes (de quelques dizaines à quelques milliers de nucléotides) que celles d'ADN (de quelques millions à quelques milliards de nucléotides).

Comme détaillé dans le paragraphe (i), il existe une grande diversité au sein de la famille des ARN.

Ces molécules d'ARN simple brin se replient sur elles-même afin de former une structure intramoléculaire plus stable et plus compacte que leur structure primaire linéaire. Cette structure repose sur la formation de liaisons internes entre bases complémentaires (A-U, G-C et parfois G-U). L'ensemble de ces appariements forme la structure 3D de l'ARN. Ces liaisons sont à l'origine de la conformation finale de la molécule d'ARN selon sa structure tertiaire. Ces conformations spatiales sont cependant dépendantes des conditions physico-chimiques de leur environnement et

20. Certains virus présentent cependant un génome à ARN sous une forme double brin, de même, les ARN impliqués dans la constitution de la petite sous-unité du ribosomes sont double brins

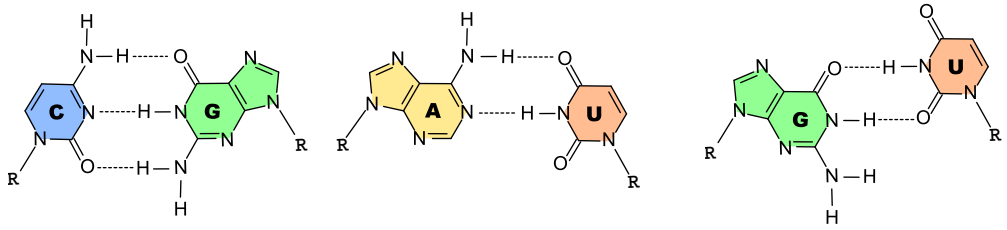
particulièrement de sa concentration en cations divalents. En effet, ceux-ci en interagissant avec les groupements phosphates du squelette de la molécule font écran à la répulsion électrostatique entre les charges de ces mêmes phosphates.

(ii) Structure Primaire de l'ARN

L'ARN est un polyribonucléotide, c'est-à-dire un polymère de ribonucléotides pris parmi quatre ribonucléotides différents : l'Adénine (A), la Guanine (G), l'Uracile (U) et la Cytosine (C) (voir Figure 1.9). Ces ribonucléotides sont reliés entre eux par des liaisons covalentes, des liaisons $3' - 5'$ phosphodiester (voir Figure 1.21). L'ordre de succession des ribonucléotides porte l'information contenue dans l'ARN en déterminant, entre autres, sa conformation spatiale. Cette structure monocaténaire, à un seul brin, est définie par une chaîne vectorisée dont le sens conventionnel est $5' \rightarrow 3'$. Cette structure constitue la structure primaire d'un ARN.

(iii) Structure Secondaire de l'ARN

Une simplification de la structure 3D des ARN est la structure secondaire où toutes les liaisons sont représentées dans le plan [21]. Ces appariements sont des liaisons hydrogènes établies entre certains nucléotides. On appelle liaisons de Watson et Crick les liaisons rapprochant A avec U et C avec G (voir Figure 1.22.a). On note également la présence de liaisons hydrogènes entre G et U, on nomme cet appariement *liaisons de Wobble* (voir Figure 1.22.b).



(a) Bases complémentaires classiques *(b) Liaison de Wobble*

FIGURE *1.22 – Différents appariements impliqués dans la structure secondaire des ARN.*

On appelle *hélice* les régions formées par l'appariement de bases successives (voir Figure 1.23). Ces structures peuvent se former grâce à des séquences répétées inversées. Ces régions s'apparient donc de manière antiparallèle afin de former un double brin localement. On observe alors que les bases non appariées entre les deux segments, soit au bout de l'hélice, forment une *boucle* ou plus précisément une *boucle*

21. Cette représentation n'existe pas dans la cellule, c'est une représentation simplifiée qu'il est possible d'étudier facilement.

terminale (voir la Figure 1.23). Entre deux hélices successives les bases non appariées constituent une *boucle interne* (voir Figure 1.23) connectant alors les deux hélices entre elles. De plus, au coeur d'une hélice, sur l'un de ses brins, gauche ou droit, certaines bases peuvent ne pas participer à l'hélice sans affecter l'appariement des bases environnantes, on parle alors de renflement, gauche ou droit (sur les tiges 1 et 3 de la Figure 1.23). Si plusieurs hélices sont reliées entre elles, la boucle interne constituant le point de branchement de chacune des hélices est appelée boucle multiple (voir la boucle multiple centrale de la Figure 1.23). Enfin, une succession d'hélices terminée par une boucle terminale est nommée *tige*. Cette structure en tige-boucle de l'ARN porte aussi le nom de structure en « épingle à cheveux » lorsque uniquement une seule hélice constitue la structure. Outre cette structure, d'autres ont été répertoriées.

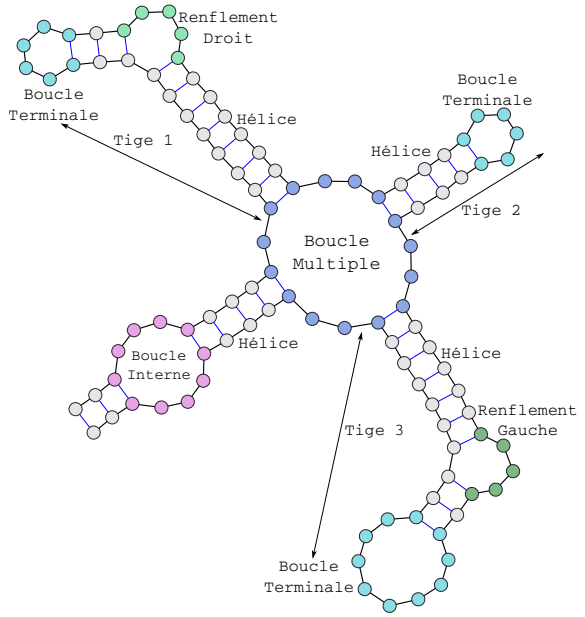


Figure *1.23 – Les différentes structures composant la structure secondaire de l'ARN.*

Au sein d'une même molécule d'ARN plusieurs régions complémentaires peuvent être identifiées et intervenir dans la conformation finale de la molécule. Ainsi, selon l'appariement de ces différentes régions, on observe des éléments topologiques variables.

Il est important de noter que la structure secondaire de l'ARN dépend des conditions physico-chimiques de son environnement (pH et forces ioniques). Une même molécule d'ARN peut ainsi adopter des conformations alternatives (Saffarian et al., 2014) selon l'environnement dans lequel elle évolue ou encore en fonction des liaisons qu'elle établie avec son ligand.

(iv) Structure Tertiaire de l'ARN

En complément de la structure secondaire, l'ARN peut adopter une conformation spatiale plus compacte ou structure tertiaire. Cette structure relève d'un niveau de repliement supplémentaire à la structure secondaire.

Celle-ci comprend de nouveaux appariements faisant toujours intervenir des liaisons hydrogènes mais qui peuvent être autres que des types Watson-Crick et Wobble, telles que des interactions base-ribose. On recense ainsi plus de 150 types d'appariements entre bases qui ont été regroupés en douze grandes familles. Cette nomenclature systématique de toutes les interactions a été proposée par Éric Westhof et ses collaborateurs (Leontis and Westhof, 2002). Cette classification repose entre autres sur la face des bases impliquées dans l'interaction (par exemple un nucléotide de type purine comporte trois faces permettant des liaisons hydrogènes).

Dans la structure tertiaire, des interactions longues distances souvent localisées dans des boucles permettent la stabilisation de la structure. Parmi ces interactions les plus inventoriées sont les pseudonoeuds formés par l'interaction d'une boucle avec une région située au delà de la tige qui la délimite.

Il est intéressant de s'attarder sur la composition de la structure des ARN puisque elle conditionne l'activité de l'ARN en question. On observe une meilleure conservation de la structure des ARN que de leur séquence puisque la structure 3D porte la fonction de l'ARN. Les ARN peuvent être classés en familles. Pour réaliser cette classification, reposant sur la fonction des ARN, la séquence et la structure entrent en compte. On remarque d'ailleurs que pour de nombreuses familles la structure est très bien conservée, comme chez les ARNt, et que seuls certains nucléotides le sont. La structure de l'ARN et plus précisément son squelette, qui peut être décomposé selon les différents motifs structuraux (tiges, boucles ou encore boucles multiples), a donc une place primordiale dans la compréhension de la fonction de l'ARN.

Afin de recenser l'ensemble des données recueillies sur les séquences ARN par la communauté mondiale, des bases de données dédiées à ces ARN ont été créées.

(v) La Rfam

La *Rfam*(Griffiths-Jones et al., 2003) est une base de données libre d'accès originellement développée par le « Wellcome Trust Sanger Institute ». Elle est actuellement maintenue à jour par l'EBI (« European Bioinformatics Institute ») et est disponible à l'adresse suivante : `http://rfam.sanger.ac.uk`. Elle recense des informations relatives aux ARNnc. Les ARNnc appartenant à une même famille présentent souvent une structure secondaire conservée sans pour autant présenter de fortes similitudes en structure primaire. L'organisation de la Rfam divise les ARNnc en familles où chaque famille est représentée par un alignement multiple des séquences qui la composent, une structure secondaire prédite et un modèle de covariance. La librairie de modèles de covariance ainsi créée peut être utilisée afin d'identifier des ARNnc inconnus homologues à des ARNnc présents dans la Rfam grâce au filtre *infernal* (Nawrocki and Eddy, 2013).

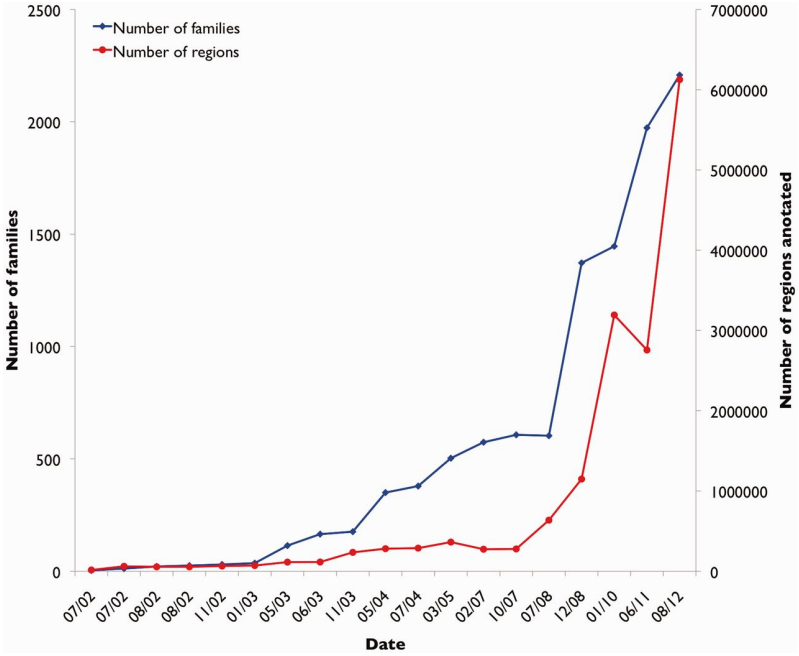


Figure 1.24 – *Évolution de la taille, en nombre de familles et en nombre de régions annotées de séquences, de la base de données Rfam à chaque nouvelle version (Burge and al., 2013). Les dates sont au format mois/année.*

En juillet 2002, la Rfam donnait accès à 15255 ARNnc classés en quatre familles, et aujourd'hui, ses utilisateurs ont accès à 19623515 séquences d'ARNnc organisés en 2450 familles recensées. Cette base de données est régulièrement entretenue puisque entre 2002 et 2014 pas moins de 19 mises à jours ont été opérées.

1.3.2 Les Pseudogènes

(i) Définition

On désigne par pseudogène les séquences géniques, codantes ou non codantes, ayant subi des altérations génétiques entraînant une incapacité à conduire à l'expression de son médiateur cellulaire et originellement dites inactives (ou non fonctionnelles) au sein d'un génome. Ces séquences ont longtemps été considérées comme des « gènes fossiles » ou des « gènes poubelles » associés aux séquences non codantes du génome (Balakirev and Ayala, 2003). Les connaissances actuelles ont montré que tout gène ne code pas nécessairement pour une protéine et les pseudogènes peuvent ainsi avoir une fonction au sein de la cellule (Balakirev and Ayala, 2003). Les pseudogènes sont donc des gènes ayant subi des altérations modifiant leur fonction d'origine et peuvent avoir une nouvelle fonction. Il est apparu suite à de récentes études que les pseudogènes pouvaient être impliqués dans la régulation de fonctions biologiques. En effet, non traduits, ils peuvent être transcrit en ARNnc qui jouent alors le rôle

de médiateur cellulaire (Hirotsune et al., 2003; Deroin, 2010).

Espèce	Taille (Mb)	Gènes Codants	Nombre de pseudogènes
Homme	3 272	22 286	16 946
Souris	1 492	6 070	7 562
Leuconostoc lactis	1,7	1686	17
Lactobacillus plantarum	3,3	3 124	23
Geobacillus thermodenitrificans	3,4	3 374	54
Mycobacterium leprae	3,3	2 770	1,116
Oenococcus oeni	1,8	1 691	122

TABLE *1.2 – Variation du nombre de pseudogènes dans diverses espèces.*[22]

Comme on peut l'observer dans la Table 1.2 (voir également l'Annexe 7), il n'existe pas de lien direct entre le nombre de pseudogènes identifiés et la taille du génome ou le nombre de gènes identifiés que l'organisme soit eucaryote ou procaryote. On remarque néanmoins que, de manière générale, la proportion de pseudogènes est plus importante chez les eucaryotes et que, ponctuellement, elle peut être élevée dans les génomes procaryotes d'organismes localisés dans des environnements potentiellement stressants.

Les analyses de génomes, et plus précisément de génomes d'espèces proches, ont mis en évidence que les pseudogènes peuvent partager des segments de séquence plus ou moins important avec des gènes du génome auquel ils appartiennent ou avec d'autres espèces (Sudbrak et al., 2003). Ceci soulève dans un premier temps la notion de famille pseudogénique et de pseudogènes unitaires. Dans un second temps, cela soulève également le problème de l'origine de ces pseudogènes. En effet, les pseudogènes pourraient être une des conséquences de l'adaptation de l'organisme à son environnement ou résulter de l'évolution de l'organisme. Ce phénomène de pseudogénisation correspond à l'accumulation d'altérations délétères au sein d'une séquence génique ou suite à des événements de remaniements. Par exemple dans le cas de duplications, l'existence d'un duplicata fonctionnel du pseudogène peut impacter la pression sélective sur la copie altérée.

(ii) Classes de Pseudogènes

Les pseudogènes proviennent de divers mécanismes biologiques liés à l'évolution des séquences génomiques. Il est possible, selon la nature des mécanismes évolutifs, de classer les pseudogènes en trois principales classes.

22. A partir des données de `www.ensembl.org`, `http://www.ncbi.nlm.nih.gov/` et `www.pseudogene.org`.

Les pseudogènes unitaires On désigne par pseudogènes unitaires les séquences pseudogéniques ne présentant aucun paralogue (Zhang et al., 2010) (voir Figure 1.20) dans le génome auquel elles appartiennent.

Ils proviennent de séquences géniques, suite à l'apparition d'altérations délétères. Structurellement, ces pseudogènes présentent toujours la structure du gène d'origine, soit hypothétiquement une structure en intron-exon dans le cas des eucaryotes. L'histoire évolutive de la séquence est importante afin de comprendre l'origine et la classification d'un pseudogène.

Certains pseudogènes peuvent être le résultat d'un transfert horizontal chez les bactéries. Ces gènes ainsi que ceux au milieu desquels ils ont été insérés peuvent alors être pseudogénisés.

Les pseudogènes dupliqués Les pseudogènes dupliqués proviennent de deux processus : les duplications en tandem et des *enjambements*[23] (Mighell et al., 2000). Ces gènes dupliqués perdent leur capacité à être traduit si, par exemple, le promoteur n'a pas été dupliqué. Le gène dupliqué peut subir des mutations invalidantes telles que des déphasages, ou « frameshift », ou l'apparition de codons stop prématurés.

Les pseudogènes retrotransposés Les pseudogènes retrotransposés ou « preprocessed pseudogènes » (notés PP) sont créés lorsque un ARNm est soumis à une transcription inverse qui synthétise une séquence ADN codante (ou ADNc). Cette séquence d'ADN est alors intégrée dans le génome à un nouvel emplacement. Cette rétrotransposition d'ARN est rendue possible grâce à une enzyme, la transcriptase inverse qui permet l'encodage inverse de l'ARNm en ADNc. Par conséquent, chez les eucaryotes, de telles séquences ne contiennent pas d'introns (Maestre et al., 1995; D'Errico et al., 2004). Les autres caractéristiques communes des PP sont la présence dans leur séquence génique d'une queue polyA et de répétitions directes présentes à chaque extrémité des pseudogènes (Maestre et al., 1995; D'Errico et al., 2004). L'activité transcriptionnelle du PP dépend de la localisation de son intégration. En effet, si il est inséré à proximité d'un autre promoteur il pourra en profiter (Zheng et al., 2007). Suite à l'analyse du génome humain il a été observé que l'ensemble des PP a été généré à partir de seulement 10% des gènes codants (Ohshima et al., 2003; Zhang et al., 2003). Ainsi les gènes les plus fortement exprimés sont davantage susceptibles de produire des PP, par exemple les gènes codants pour les protéines ribosomales représentent environ 20% des PP humains (Zhang et al., 2002).

(iii) Évolution et Conservation des Pseudogènes

Les pseudogènes ont longtemps été considérés comme des « séquences neutres », c'est-à-dire des séquences dans lesquelles les mutations s'accumulent et ne sont pas soumises à la pression de sélection (Li et al., 1981). Cependant ce principe repose

23. on appelle enjambement ou « crossing-over » le phénomène génétique au cours duquel les chromosomes échangent des fragments de leur chromatide

sur l'hypothèse selon laquelle les pseudogènes sont fonctionnellement inertes, ce qui ne reflète pas la réalité. De récentes études ont montré que certains pseudogènes peuvent être fonctionnellement actifs (Deroin, 2010; Fujii et al., 1999). L'étude de l'évolution et de la conservation du pseudome, soit de l'ensemble des pseudogènes d'un génome, pourrait alors apporter des compléments d'information quant à leur rôle fonctionnel et leurs mécanismes d'action au sein de la cellule.

Comme nous venons de le voir les pseudogènes proviennent de l'évolution des gènes du génome et de l'accumulation progressive d'altérations sur leur séquence. Ils sont ainsi les vestiges de l'évolution de ce génome, et potentiellement le témoignage de sa composition fonctionnelle à un instant t passé.

De plus, il a été observé que des pseudogènes étaient conservés entre plusieurs espèces. On citera par exemple l'étude des pseudogènes à l'échelle du génome humain complet faite par Svensson *et al.* (Svensson et al., 2006) via une approche comparative entre l'homme et la souris. Étonnamment parmi les pseudogènes analysés, une trentaine est présente dans les deux organismes et les séquences sont peu divergentes. Ceci implique à la fois que la pseudogénisation a eu lieu avant la spéciation entre ces deux espèces et que ces séquences pourraient avoir un rôle fonctionnel au sein des organismes qui les possèdent.

(iv) Transcription des Pseudogènes

La plupart des pseudogènes perdent leur capacité à être transcrits suite à une altération dans leur promoteur ou à leur intégration dans une région silencieuse du génome (dans le cas des PP). De plus, de part leur similitude en séquence avec les gènes fonctionnels dont ils proviennent, il peut s'avérer complexe de mesurer spécifiquement le taux de transcription de ces seuls pseudogènes (Harper et al., 2003). Cependant, à ce jour il existe de nombreux exemples pour lesquels il est démontré que les pseudogènes sont transcrits et ont un rôle fonctionnel via leur ARNnc, comme par exemple le pseudogène du suppresseur de tumeur PTEN (Fujii et al., 1999). Il a pu être démontré que l'expression du gène suppresseur de tumeur PTEN est contrôlée par l'ARNnc exprimé par le pseudogène de PTEN.

L'étude des profils d'expression (transcription) permet d'obtenir des indications quant au potentiel fonctionnel des séquences. Il a été observé que de nombreux ARNnc présentaient des profils d'expressions tissus spécifiques et possédaient un rôle fonctionnel, parmi ces ARNnc on peut citer les ARN antisens (Katayama et al., 2005), les micro ARN (ou miARN). Il est également intéressant de relever qu'il existe de nombreux exemples pour lesquels l'expression spatio-temporelle du pseudogène diffère de celle du gène dont il est issu (Elliman et al., 2006).

L'expression des pseudogènes varie selon les conditions physiologiques de l'organisme. Par exemple, chez *Mycobacterium leprae*, l'organisme à l'origine de la lèpre, l'expression des pseudogènes fluctue au cours du processus d'infection (Suzuki et al., 2006).

La transcription des pseudogènes dépend en partie du promoteur dont il dépend. Certains possèdent leur propre promoteur alors que d'autres profitent du promoteur d'un gène situé à proximité (Vinckenbosch et al., 2006). En outre, il est évident que la localisation de l'intégration des PP dans la séquence génomique est importante. L'expression différentielle entre le pseudogène et son gène parent ne reflète pas nécessairement une activité mais résulte de l'utilisation d'un promoteur différent. Avec une telle affirmation, les pseudogènes devraient avoir un impact neutre en terme de gain en aptitudes (évolution) et auraient même tendance à disparaître. Or l'analyse de la transcription des pseudogènes chez les primates montre un taux de conservation au cours de l'évolution de 50%. Il est également intéressant de remarquer que dans cette étude le taux de conservation tend à diminuer avec l'éloignement des espèces étudiées (Khachane and Harrison, 2009). Cette observation de la conservation au cours des millénaires de certains pseudogènes suggère un rôle fonctionnel de ces séquences dans les cellules.

(v) Des Pseudogènes Fonctionnels ?

De nombreux organismes semblent conserver les pseudogènes malgré les contraintes évolutives. Ce phénomène est d'ailleurs majoritairement observé chez les organismes pluricellulaires (Kuo and Ochman, 2010). Quels bénéfices ces organismes retirent-ils de ces pseudogènes qui ont potentiellement perdu leur capacité de coder une protéine fonctionnelle ? Dans un premier temps, ces séquences peuvent être un apport comme source de diversité génétique, par exemple pour la formation d'antigènes et d'anticorps (Balakirev and Ayala, 2003). Chez les bactéries, en revanche, on observe un faible nombre de pseudogènes. Une des raisons peut être liée au principe de compaction. C'est en tant que régulateur via leur ARN non codant (ou ARNnc) que les pseudogènes présentent leur plus important potentiel. Le rôle régulateur des ARNnc a été démontré dans certains processus cellulaires. Les ARNnc synthétisés à partir de la séquence de pseudogènes sont alors apparus comme participant à certains mécanismes de contrôle de fonctions géniques.

Les transcrits antisens des pseudogènes peuvent avoir un rôle de régulateur de leur gène parent en s'appariant pour former un ARN double brin avec le transcrit sens du gène. Par exemple, le « knockdown [24] » à l'aide d'un ARN antisens du pseudogène de Oct4 conduit à une augmentation de l'expression de Oct4 (Hawkins and Morris, 2010).

La stabilité d'un ARNm dépend à la fois de l'activité de ses séquences en *cis* et de leur interaction avec leurs facteurs *trans* (Ross, 1996). Ainsi si un pseudogène et son gène codant parent sont soumis aux mêmes séquences en *cis*, les deux séquences entrent alors en compétition pour les mêmes facteurs en *trans*. La stabilité de l'ARNm est alors altérée et par conséquent sa traduction en protéine aussi.

24. Le « knockdown » réfère à une technique expérimentale par laquelle on réduit l'expression d'un ou plusieurs gènes ; soit par modification génétique soit par traitement avec une séquence complémentaire à l'ARNm

(vi) Bases de Données et Méthode de Détection des Pseudogènes

Depuis la découverte de ces nouveaux objets géniques, des méthodes de détection automatique ont été élaborées et de nouvelles bases de données ont été créées pour répertorier ces nouveaux objets géniques détectés.

Détection des pseudogènes La plupart des méthodes de détection des pseudogènes se basent sur la recherche d'homologie afin d'identifier des régions pseudogéniques potentielles. Au cours de cette approche on compare un génome (ou un chromosome) d'intérêt avec un jeu de protéines connues (*ENSEMBL* (Hubbard et al., 2002), *UniProt* (Consortium et al., 2008),...). On compare des séquences nucléotidiques avec des séquences peptidiques, les séquences nucléotidiques sont alors traduite dans les six phases de lecture grâce par exemple à un tBLASTn (Altschul and al, 1990). Enfin, la détection des séquences pseudogéniques se termine par une étape de filtrage permettant d'identifier les événements géniques à l'origine de l'état altéré de la séquence.

Depuis 2001, plusieurs méthodes de détection des pseudogènes ont été élaborées mais dans la plupart des cas dédiées aux eucaryotes. Harrison et al. (2001) ont développé un outil d'annotation des pseudogènes à partir du génome de *Caenorhabditis elegans* puis au cours d'une seconde étude sur les chromosomes 21 et 22 du génome humain (Harrison et al., 2002). D'autres méthodes ont été développées telles que *PFINDER* (van Baren and Brent, 2006) qui après l'établissement d'un modèle génique se base sur des méthodes de localisation des introns et de conservation de la synténie[25]. *Pseudogene Finder* (*PSF*) a été développé avec 44 séquences du projet ENCODE (Consortium et al., 2004) à partir de l'alignement de protéines sur ces séquences dont les localisations ne correspondent pas à des gènes identifiés et qui ont un degré de similarité suffisamment élevé. Zhang et al. (2006) ont mis en place une méthode automatique de détection des pseudogènes basée sur la notion d'homologie (tBLASTn).

Bases de données Quelques bases de données regroupant les pseudogènes identifiés ont été mises en place. *HOPPSIGEN* (Adel et al., 2005) est une base de données nucléiques de pseudogènes rétrotransposés identifiés chez l'homme et la souris et développée par le Pôle Bioinformatique Lyonnais. *Pseudogene.org* (Karro et al., 2007) est une base de données dédiée aux pseudogènes d'environ une centaine de génomes et maintenue par le Yale Gerstein Group.

Tout comme les méthodes d'identification, les bases de données pseudogéniques sont essentiellement dédiées aux organismes eucaryotes.

25. La synténie, ou colinéarité, fait référence à la conservation de l'organisation des loci présents sur un chromosome dans deux espèces différentes.

Pour conclure cette description, les pseudogènes sont des structures géniques de plus en plus étudiées mais il faut rester prudent quant aux interprétations des résultats expérimentaux. En effet, dans certains cas, un hypothétique pseudogène peut en réalité coder pour une protéine tronquée. Néanmoins, de plus en plus d'expériences tendent à démontrer que les pseudogènes ont une fonction régulatrice dans la cellule via leur ARNnc ou interviennent dans la régulation de l'expression de gènes qui leur sont similaires (Pink et al., 2011). En outre, tous les pseudogènes ne présentent pas de fonction biologique et seuls ceux qui présentent un bénéfice pour la cellule tendent à être conservés.

Il semble intéressant d'aborder l'étude de ce répertoire pseudogénique, de ses caractéristiques, ainsi que son évolution afin d'en extraire des règles qui régissent ces objets. En effet, outre leur implication dans les fonctions cellulaires, de part leur séquence proche de celle des gènes codants, les pseudogènes leurrent les algorithmes d'annotation des séquences génomiques.

Bibliographie

Adel, K., Laurent, D., and Dominique, M. (2005). Hoppsigen : a database of human and mouse processed pseudogenes. *Nucleic acids research*, 33(suppl 1) :D59–D66.

Altschul, S. and al (1990). Basic local alignment search tool. *Journal of molecular biology*, 215 :403–410.

Balakirev, E. S. and Ayala, F. J. (2003). Pseudogenes : are they junk or functional dna ? *Annual review of genetics*, 37(1) :123–151.

Buckingham, S. et al. (2003). The major world of micrornas. *Nature*, 4.

Burge, S. and al. (2013). Rfam 11.0 : 10 years of rna families. *Nucleic Acids Research*, pages D226–D232.

Collins, F. and al. (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431 :931–945.

Consortium, E. P. et al. (2004). The encode (encyclopedia of dna elements) project. *Science*, 306(5696) :636–640.

Consortium, U. et al. (2008). The universal protein resource (uniprot). *Nucleic acids research*, 36(suppl 1) :D190–D195.

Crick, F. and Watson, J. (1953). Molecular structure of nucleic acids. *Nature*, 171(4356) :737–738.

Deroin, P. (2010). Des pseudogenes pas si pseudo. *Biofutur*, (313).

D'Errico, I., Gadaleta, G., and Saccone, C. (2004). Pseudogenes in metazoa : origin and features. *Briefings in functional genomics & proteomics*, 3(2) :157–167.

Dufey, F. (1986). *Biologie cellulaire*. Kinshasa.

Durrens, P., Nikolski, M., and Sherman, D. (2008). Fusion and fission of genes define a metric between fungal genomes. *PLoS computational biology*, 4(10) :e1000200.

Elliman, S. J., Wu, I., and Kemp, D. M. (2006). Adult tissue-specific expression of a dppa3-derived retrogene represents a postnatal transcript of pluripotent cell origin. *Journal of Biological Chemistry*, 281(1) :16–19.

Fujii, G. H., Morimoto, A. M., Berson, A. E., and Bolen, J. B. (1999). Transcriptional analysis of the pten/mmac1 pseudogene, psipten. *Oncogene*, 18(9) :1765–1769.

Giegé, R., Frugier, M., and Rudinger, J. (1998). trna mimics. *Current opinion in structural biology*, 8(3) :286–293.

Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., and Eddy, S. R. (2003). Rfam : an rna family database. *Nucleic acids research*, 31(1) :439–441.

Hacker, J. and Carniel, E. (2001). Ecological fitness, genomic islands and bacterial pathogenicity. *EMBO reports*, 2(5) :376–381.

Harper, L. V., Hilton, A. C., and Jones, A. F. (2003). Rt-pcr for the pseudogene-free amplification of the glyceraldehyde-3-phosphate dehydrogenase gene (< i> gapd</i>). *Molecular and cellular probes*, 17(5) :261–265.

Harrison, P., Echols, N., and Gerstein, M. (2001). Digging for dead genes : an analysis of the characteristics of the pseudogene population in the caenorhabditis elegans genome. *Nucleic acids research*, 29(3) :818–830.

Harrison, P., Hegyi, H., Balasubramanian, S., Luscombe, N., Bertone, P., Echols, N., Johnson, T., and Gerstein, M. (2002). Molecular fossils in the human genome : identification and analysis of the pseudogenes in chromosomes 21 and 22. *Genome research*, 12(2) :272–280.

Hawkins, P. and Morris, K. (2010). Transcriptional regulation of oct4 by a long non-coding rna antisense to oct4-pseudogene 5. *Transcription*, 1(3) :165–175.

Hirotsune, S., Yoshida, N., Chen, A., Garrett, L., Sugiyama, F., Takahashi, S., Yagami, K., Wynshaw-Boris, A., and Yoshiki, A. (2003). An expressed pseudogene regulates the messenger-rna stability of its homologous coding gene. *Nature*, 423(6935) :91–96.

Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L., Cox, T., Cuff, J., Curwen, V., Down, T., et al. (2002). The ensembl genome database project. *Nucleic acids research*, 30(1) :38–41.

Jacq, C Miller, J. and Brownlee, G. (1977). A pseudogene structure in 5s dna of xenopus laevis. *Cell*, 12 :109–120.

Kamalika, S. and Tapash, C. (2013). Pseudogenes and their composers : delving in the 'debris' of human genome. *Functional Genomics*, 12 :536–547.

Karro, J., Yan, Y., Zheng, D., Zhang, Z., Carriero, N., Cayting, P., Harrrison, P., and Gerstein, M. (2007). Pseudogene. org : a comprehensive database and comparison platform for pseudogene annotation. *Nucleic acids research*, 35(suppl 1) :D55–D60.

Katayama, S., Tomaru, Y., Kasukawa, T., Waki, K., Nakanishi, M., Nakamura, M., Nishida, H., Yap, C., Suzuki, M., Kawai, J., et al. (2005). Antisense transcription in the mammalian transcriptome. *Science*, 309(5740) :1564–1566.

Khachane, A. N. and Harrison, P. M. (2009). Assessing the genomic evidence for conserved transcribed pseudogenes under selection. *BMC genomics*, 10(1) :435.

Kuo, C.-H. and Ochman, H. (2010). The extinction dynamics of bacterial pseudogenes. *PLoS genetics*, 6(8) :e1001050.

Leontis, NB Stombaugh, J. and Westhof, E. (2002). The non-watson-crick base pairs and their associated isostericity matrices. *Nucleic Acid Research*, 30 :3497–3531.

Li, W.-H., Gojobori, T., Nei, M., et al. (1981). Pseudogenes as a paradigm of neutral evolution. *Nature*, 292(5820) :237–239.

Maestre, J., Tchenio, T., Dhellin, O., and Heidmann, T. (1995). mrna retroposition in human cells : processed pseudogene formation. *The EMBO journal*, 14(24) :6333.

Mallick, B. and Ghosh, Z. (2012). *Regulatory RNAs : Basics, Methods and Applications*. Springer.

Mighell, A., Smith, N., Robinson, P., and Markham, A. (2000). Vertebrate pseudogenes. *FEBS letters*, 468(2) :109–114.

Nawrocki, E. and Eddy, S. (2013). Infernal 1.1 : 100-fold faster rna homology searches. *Bioinformatics*, 29 :2933–2935.

Noller, H. F., Hoffarth, V., and Zimniak, L. (1992). Unusual resistance of peptidyl transferase to protein extraction procedures. *Science*, 256(5062) :1416–1419.

Ohshima, K., Hattori, M., Yada, T., Gojobori, T., Sakaki, Y., and Okada, N. (2003). Whole-genome screening indicates a possible burst of formation of processed pseudogenes and alu repeats by particular li subfamilies in ancestral primates. *Genome biology*, 4(11) :R74–R74.

Pink, R., Wicks, K., Caley, D., Punch, E., Jacobs, L., and Carter, D. (2011). Pseudogenes : pseudo-functional or key regulators in health and disease ? *Rna*, 17(5) :792–798.

Ross, J. (1996). Control of messenger rna stability in higher eukaryotes. *Trends in Genetics*, 12(5) :171–175.

Saffarian, A., Giraud, M., and Touzet, H. (2014). Searching for alternate rna structures in genomic sequences. In *Proceedings of 1st Workshop on Computational Methods for Structural RNAs (CMSR'14)*, volume 1, pages 13–24.

Sanger, F., Air, G., Barrell, B., Brown, N., Coulson, A., Fiddes, C., Hutchison, C., PM, S., and Smith, M. (1977a). Nucleotide sequence of bacteriophage phi x174 dna. *Nature*, 265 :687–695.

Sanger, F., Nicklen, S., and Coulson, A. (1977b). Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12) :5463–5467.

Schwann, T. (1837). Vorlaufige mitteilung betreffend versuche uber die weingahrung und faulniss. *Annalen der Physik und Chemie*, XLI :184–193.

Sudbrak, R., Reinhardt, R., Hennig, S., Lehrach, H., Günther, E., and Walter, L. (2003). Comparative and evolutionary analysis of the rhesus macaque extended mhc class ii region. *Immunogenetics*, 54(10) :699–704.

Suzuki, K., Nakata, N., Bang, P. D., Ishii, N., and Makino, M. (2006). High-level expression of pseudogenes in mycobacterium leprae. *FEMS microbiology letters*, 259(2) :208–214.

Svensson, Ö., Arvestad, L., and Lagergren, J. (2006). Genome-wide survey for biologically functional pseudogenes. *PLoS computational biology*, 2(5) :e46.

van Baren, M. and Brent, M. (2006). Iterative gene prediction and pseudogene removal improves genome annotation. *Genome research*, 16(5) :678–685.

Vinckenbosch, N., Dupanloup, I., and Kaessmann, H. (2006). Evolutionary fate of retroposed gene copies in the human genome. *Proceedings of the National Academy of Sciences of the United States of America*, 103(9) :3220–3225.

Wajcman, H., Kiger, L., and Marden, M. (2009). Structure and function evolution in the superfamily of globins. *Comptes rendus biologies*, 332(2) :273–282.

Williams, A., Spilianakis, C., and Flavell, R. (2010). Interchromosomal association and gene regulation in trans. *Trends in genetics*, 26(4) :188–197.

Wong, A., Ruppert, J., Eggleston, J., Baylin, S., Vogelstein, B., et al. (1986). Gene amplification of c-myc and n-myc in small cell carcinoma of the lung. *Science*, 233(4762) :461–464.

Yan, B. and Wang, Z. (2012). Long noncoding rna : its physiological and pathological roles. *DNA and cell biology*, 31(S1) :S–34.

Zhang, Z., Carriero, N., Zheng, D., Karro, J., Harrison, P., and Gerstein, M. (2006). Pseudopipe : an automated pseudogene identification pipeline. *Bioinformatics*, 22(12) :1437–1439.

Zhang, Z., Harrison, P., and Gerstein, M. (2002). Identification and analysis of over 2000 ribosomal protein pseudogenes in the human genome. *Genome research*, 12(10) :1466–1482.

Zhang, Z., Harrison, P. M., Liu, Y., and Gerstein, M. (2003). Millions of years of evolution preserved : a comprehensive catalog of the processed pseudogenes in the human genome. *Genome research*, 13(12) :2541–2558.

Zhang, Z. D., Frankish, A., Hunt, T., Harrow, J., and Gerstein, M. (2010). Identification and analysis of unitary pseudogenes : historic and contemporary gene losses in humans and other primates. *Genome Biol*, 11(3) :R26.

Zheng, D., Frankish, A., Baertsch, R., Kapranov, P., Reymond, A., Choo, S. W., Lu, Y., Denoeud, F., Antonarakis, S. E., Snyder, M., et al. (2007). Pseudogenes in the encode regions : consensus annotation, analysis of transcription, and evolution. *Genome research*, 17(6) :839–851.

Chapitre 2

Comparaisons d'Objets Biologiques

Nous venons de voir qu'à ce jour il est possible de séquencer les génomes tout comme les protéines de divers organismes. Cependant, même une fois séquencés cela ne donne pas systématiquement des informations sur la fonction ou sur la structure de ces séquences. De plus, nous avons également vu qu'au cours de l'évolution (se référer à la Section 1.2), les séquences nucléiques sont soumises à des remaniements ou des transformations. Il est alors intéressant d'établir le parcours évolutif de ces séquences ainsi que leurs liens de parenté afin d'établir une phylogénie entre celles-ci.

C'est dans cette optique qu'est apparue la génomique comparative qui consiste, entre autres, à quantifier la similitude entre deux séquences, c'est-à-dire à extraire de l'information des ressemblances et dissemblances observables entre plusieurs séquences. Cette étude comparative permet de transférer des informations connues d'une séquence à une autre et permettent ainsi d'identifier des sites fonctionnels, de prédire la fonction ou la structure secondaire, d'inférer une séquence,...

Au cours de ce deuxième chapitre nous nous attacherons à introduire et définir l'ensemble des concepts sur lesquels nos travaux se basent. Nous présenterons les bases algorithmiques à nos travaux, à savoir la comparaison et l'alignement de séquences suivis du principe de chaînage. Nous terminerons par une description des filtres FastA et BLAST qui présentent une architecture similaire à nos travaux.

2.1 Comparaison de Séquences

Dans cette section, nous allons décrire l'algorithme d'alignement de séquences permettant une comparaison de deux séquences. Avant cela nous introduirons les principales notations qui seront utiles à la compréhension de l'ensemble de ce manuscrit. Nous aborderons ensuite la définition du problème d'alignement global de séquences et sa déclinaison local. L'ensemble de ces points nous mèneront à la question relative au chaînage.

2.1.1 Modélisation, Notations et Définitions

Modélisation des Séquences

Commençons par donner une définition formelle d'une séquence et des propriétés qui y sont associées.

Définition 1 (Séquence)

On définit une séquence S comme la suite ordonnée de $|S|$ symboles pris dans un alphabet Σ de taille $|\Sigma|$.
Pour i compris entre 0 et $|S|-1$, on dénote par $S[i]$ le $i{+}1^{ème}$ symbole de S ou encore symbole à la position i de S.

On note qu'une séquence ADN est définie sur l'alphabet $\{A, T, C, G\}$ alors qu'une séquence ARN est définie sur l'alphabet $\{A, C, G, U\}$. [1]

On définit le facteur d'une séquence comme suit :

Définition 2 (Facteur, Préfixe et Suffixe)

Soit S une séquence de taille $|S|$ et deux entiers i et j tels que $i, j \in [0, |S|[$. On appelle facteur de la séquence S, $S[i, j]$, la séquence formée par la suite des symboles aux positions $i, i+1, \cdots, j$ de S.
Si $j < i$, alors $S[i, j]$ représente le mot vide ϵ.
Si $i = 0$, alors on parle de préfixe de longueur $j+1$.
Si $j = |S|-1$, alors on parle de suffixe à la position i.

Enfin, on appelle respectivement plus long préfixe propre, $\overrightarrow{S}$, et plus long suffixe propre, $\overleftarrow{S}$, les facteurs $S[0, |S|-2]$ et $S[1, |S|-1]$.

Alignement Global de Deux Séquences

Soit deux séquences S_1 et S_2, un alignement de ces deux séquences consiste en un ensemble d'associations d'une position de S_1 avec une position de S_2. Ces mises en correspondance respectent la relation d'ordre imposée par S_1 et S_2.

Pour chaque association, si les deux symboles sont identiques, on parle de « matchs » et s'ils sont différents, on parle alors de « mismatchs ». Enfin, pour les positions de S_1 et S_2 ne participant pas à une association, on parle de « gaps ». Ces trois mises en correspondance font référence à un phénomène génétique : la mutation.

1. On se restreindra à l'alphabet général, mais il existe un alphabet étendu.

Définition 3 (Mutations)

Soit une séquence S sur un alphabet Σ, on définit une mutation comme une modification ponctuelle d'un symbole de S. On répertorie 3 types de mutations :

- *les substitutions qui correspondent au remplacement d'un symbole i de la séquence S par un autre symbole pris dans le même alphabet $\Sigma \setminus \{i\}$.*
- *les insertions et délétions qui correspondent respectivement à l'ajout d'un symbole pris dans l'alphabet Σ ou à la suppression d'un symbole dans la séquence S. On appelle ces types de mutation « indel ».*

On définit alors un alignement global comme suit :

Définition 4 (Alignement global valide entre deux séquences)

Soit deux séquences S_1 et S_2 de tailles respectives $|S_1|$ et $|S_2|$ sur l'alphabet Σ. Un alignement global valide de S_1 et S_2 correspond à deux séquences S'_1 et S'_2 de même longueur ω (tel que $\omega \geq |S_1|$ et $\omega \geq |S_2|$) sur l'alphabet $\Sigma \cup \{" - "\}$ telles que :

- *si on supprime tous les caractères spéciaux " − " de S'_1,respectivement S'_2, on obtient S_1, respectivement S_2.*
- *il n'existe pas de position i telle que $S'_1[i] = S'_2[i] = " - "$*

Si l'on définit les fonctions f_1 et f_2 telles que $f_1(i)$ (respectivement $f_2(j)$) fournit la position de S_1 correspondant à $S'_1[i]$ (respectivement $S'_2[j]$), alors on peut définir l'alignement $S'_1S'_2$ comme l'ensemble $\mathcal{A} = \{(f_1(i), f_2(i)) \ / 0 \leq i < \omega,\ S'_1[i] \neq' -' \text{ et } S'_2[i] \neq' -'\}$ des associations entre des positions de S_1 et de S_2. On remarque alors que par construction, pour tous couples $(i, j), (k, l) \in \mathcal{A}$:

- *si $i = k$ alors $j = l$*
- *si $i < k$ alors $j < l$*
- *si $i > k$ alors $j > l$*

Score d'Alignement

Pour deux séquences données il est alors possible d'obtenir plusieurs alignements différents. C'est pourquoi on définit un score d'alignement. Celui-ci permet alors d'attribuer à chaque alignement un score qui dépend des associations réalisées. On définit le score d'alignement comme suit :

Définition 5 (Score d'alignement)

Soit l'alignement composé des deux séquences S'_1 et S'_2 de taille ω et une fonction de score $score(x, y)$ qui associe à chaque couple de symboles (x, y), sur $(\Sigma \cup \{" - "\})^2$, un score de similarité dans $\mathbb{R}$. Le score $\mathcal{S}(S'_1, S'_2)$ associé à l'alignement est défini tel que :

$$\mathcal{S}(S'_1, S'_2) = \sum_{i=0}^{\omega} score(S'_1[i], S'_2[i])$$

Il est usuel, en particulier dans le cadre de la bioinformatique, d'encoder les valeurs de la fonction de score sous la forme d'une matrice dite « matrice de substitution ».

Définition 6 (Matrice de substitution)

Soient P une matrice carrée de taille $|\Sigma|+1$, x et y deux symboles pris dans l'alphabet $\{\Sigma \cup " - "\}$ et $ord(x)$ une bijection sur $\Sigma \cup \{" - "\}$ dans $[0, |\Sigma|]$. Le score associé à la mise en correspondance de x avec y est alors donné par :

$$score(x, y) = P[ord(x)][ord(y)]$$

Afin de calculer le score d'alignement, des matrices de substitution existent pour chaque alphabet (nucléique ou protéique) et permettent de calculer ce score en associant à chaque couple de symboles ($S_1[i]$, $S_2[j]$) mis en correspondance lors de l'alignement un score de similarité. Soient a et b deux éléments pris dans l'alphabet considéré complété par le caractère de « gap » '-'. Une telle matrice recense le score élémentaire $P[a][b]$ de substitution ou d'indel de la base a par la base b. Pour les acides nucléiques on emploie principalement la matrice d'identité ou unitaire (voir la Figure 2.25). Pour chaque position de l'alignement, la matrice permet de rendre compte de l'identité des résidus et donc de leur bonne ou au contraire mauvaise association. Néanmoins ce premier critère de ressemblance ne permet pas toujours de révéler les similitudes entre séquence. En effet, la prise en compte des mutations de types insertions et délétions d'une ou plusieurs bases par l'utilisation de gap pénalisant améliore le score associé et met en avant les zones proches.

Il est important ici de noter que ce score de similarité est fonction de la portion de similitude considérée, plus cette zone est longue plus le score est élevé. De plus, la valeur de ce score peut être nuancé par les valeurs associées aux gaps et aux valeurs des scores élémentaires.

Matrices de Substitution

Les matrices nucléiques En raison de la pauvreté de l'alphabet nucléique, on recense peu de matrices nucléiques. La plus utilisée est la matrice d'identité (voir Figure 2.25). Une seconde matrice privilégiant certaines associations, les transitions devant les transversions (voir Figure 2.25), est également utilisée.

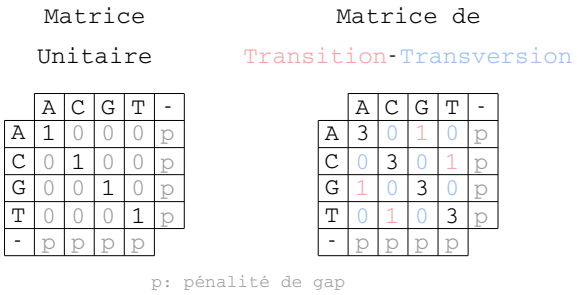


Figure 2.25 – *Les matrices nucléiques unitaire et de transversion-transition*

Les matrices protéiques La sensibilité satisfaisante obtenue par les matrices nucléiques basées sur l'identité des acides nucléiques ne l'est plus pour les matrices protéiques. En effet, si l'on tient compte du fait qu'un acide aminé peut être remplacé par un autre ayant des propriétés similaires sans pour autant altérer fondamentalement la structure ou la fonctionnalité de la protéine, un système de score classant les acides aminés en familles selon leurs affinités est plus approprié pour rendre compte des similarités entre résidus. Les matrices de score qui découlent de ce système permettent d'augmenter la fiabilité des recherches de similitudes. Dans un premier temps Fitch (Fitch, 1966) établit une matrice basée sur la dégénérescence du code génétique pour laquelle le score élémentaire rend compte du nombre commun de nucléotides dans les codons des acides aminés. Cette première matrice protéique détermine donc le nombre de mutations minimum afin de convertir un acide aminé en un autre. Depuis on distingue deux catégories de matrices, celles basées sur les substitutions d'acides aminés au cours de l'évolution et celles basées sur les caractéristiques physico-chimiques des acides aminés. De nombreuses matrices de chaque catégories ont été créées. Nous nous intéresserons, donc seulement à celles les plus couramment utilisées.

Parmi les *matrices protéiques liées à l'évolution* on distingue deux d'entre elles : les matrices de type PAM (Percent Accepted Mutations) ou matrices de mutation de Dayhoff (Dayhoff and Schwartz, 1978) et les matices de type BLOSUM (BLOcks SUbstitution Matrix) (Henikoff and Henikoff, 1992).

Les matrices de type PAM ont été déduites de l'étude de l'alignement de 71 familles de protéines (environ 1300 séquences) présentant au moins 85% de similitudes. Les alignements produits ont alors permis le calcul d'une matrice de probabilités. Cette matrice correspond à une mutation acceptée pour 100 sites au cours d'un

temps d'évolution donné, soit une substitution qui ne modifie pas significativement l'activité de la protéine : on parle de matrice 1PAM. Par multiplication de la matrice par elle même on obtient la matrice des probabilités de substitution pour des distances évolutives plus grandes : $PAM^n = nPAM$. Enfin, afin d'être plus aisément utilisables par les logiciels de comparaison qui se basent sur la similitude les matrices $nPAM$ sont transformées en matrices de similarité $PAM - n$. Les études de Swartz et Dayhoff (Dayhoff and Schwartz, 1978) portant sur ces matrices ont montré que pour distinguer les protéines apparentés de celles présentant des similitudes dues au hasard la matrice $PAM - 250$ semble optimale : elle est devenue la matrice de mutation standard de Dayhoff. Cependant cette matrice présente des inconvénients puisqu'elle considère tous les points de mutation comme équiprobables or il a été démontré que certaines positions sont plus conservées que d'autres car plus impliquées dans la fonction de la protéine. De plus les familles étudiées pour la construction de la matrice en 1978 ne sont plus représentatives de l'ensemble des familles découvertes à ce jour. C'est ainsi qu'en 1992 Jones (Jones et al., 1992) a opéré une réactualisation de cette matrice en prenant en considération 2621 familles de protéines (16130 séquences) à partir de la base de données Swissprot (Boeckmann et al., 2003).

Alors que les matrices PAM dérivent de l'alignement global de séquences protéiques proches, les matrices BLOSUM qui permettent également de rendre compte du caractère de substitution des acides aminés sont, elles, établies en observant des blocs d'acides aminés issus des protéines éloignées dans une base de plus de 2000 blocs (Henikoff and Henikoff, 1992). On obtient également les blocs par alignements multiples mais sans insertions ni délétions de courtes régions très conservées. Ces blocs permettent alors de regrouper les segments de séquence ayant un pourcentage d'identité minimum au sein du bloc puis d'en déduire les fréquences de substitution pour chaque paire d'acides aminés. Enfin la matrice BLOSUM est obtenue en calculant le logarithme de la matrice obtenue. Ainsi pour chaque pourcentage d'identité on obtient une matrice particulière, par exemple la $BLOSUM-60$ est obtenue pour un seuil de 60% d'identité.

Même si les matrices liées à l'évolution permettent un relatif regroupement des caractéristiques chimiques et structurales des acides aminés, elles ne suffisent pas pour révéler certaines caractéristiques physico-chimiques entre deux séquences peptidiques. C'est dans cette optique et en tenant compte des remplacements conservatifs que les *matrices protéiques liées aux propriétés physico-chimiques* ont été calculées. Comme souligné par le diagramme de Venn Figure 2.26 certains acides aminés ont des caractéristiques proches comme par exemple la lysine (K) et l'arginine (R) qui possèdent tous deux des chaînes latérales chargées positivement ou encore l'importance du caractère hydrophobe ou au contraire hydrophile des acides aminés dans la structure secondaire et tertiaire de la protéine. C'est pourquoi les matrices les plus courantes sont basées sur cette propriété des acides aminés avec par exemple la matrice d'hydrophobicité calculée à partir de l'énergie libre de transfert de l'eau à

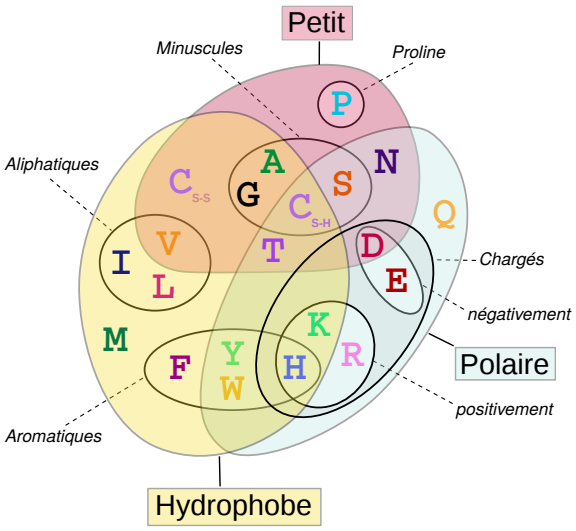


Figure 2.26 – *Diagramme de Venn présentant les acides aminés selon leurs propriétés physico-chimiques.*

l'éthanol des acides aminés (Levitt, 1976), la matrice des structures secondaires basée sur la propension d'un acide aminé à être dans une conformation donnée (Levin et al., 1986) ou encore plus récemment les matrices basées sur la comparaisons des structures tri-dimensionnelles (Risler et al., 1988).

2.1.2 Le Problème d'Alignement Global Optimal

Maintenant que nous avons introduit la notion d'alignement global valide (voir la Définition 2.1.1), nous pouvons introduire le problème d'alignement global optimal comme suit :

Définition 7 (Alignement Global Optimal)

Soient deux séquences S_1 et S_2 et $\mathcal{A} = A_0 \dots A_\ell$ la liste des $\ell+1$ alignements globaux valides de S_1 et S_2 de scores respectifs $s_0 \dots s_\ell$. On définit l'alignement optimal comme l'alignement $A_k \in \mathcal{A}$ de score maximal s_k tel que $\forall i \in [0, \ell], s_k \geq s_i$

On note que pour deux séquences données, il est possible d'obtenir plusieurs alignements de même score. Ainsi il est possible d'obtenir plusieurs alignements optimaux.

Il existe un lien fort entre l'alignement de S_1 et S_2 et les alignements entre $\overrightarrow{S_1}$ - $\overrightarrow{S_2}$, $\overrightarrow{S_1}$ - S_2 et S_1 - $\overrightarrow{S_2}$ où $\overrightarrow{S_1}$ et $\overrightarrow{S_2}$ sont les plus longs préfixes propres de S_1 et S_2.

En effet à partir de ces alignements il est possible de construire un alignement de S_1 et S_2 pour lequel :

- le dernier symbole de S_1 et le dernier symbole de S_2 sont associés.

- le dernier symbole de S_1 n'est pas associé au dernier symbole de S_2.
- le dernier symbole de S_2 n'est pas associé au dernier symbole de S_1.

Il est possible de montrer que ces trois possibilités couvrent l'ensemble des alignements possibles entre S_1 et S_2. Si on intègre le score associé à cette dernière opération (match, mismatch ou indel) aux scores des alignements connus, alors on peut démontrer que les solutions de scores maximaux correspondent à des alignements valides et optimaux de S_1 et S_2.

De ce raisonnement, on peut déduire la formule de récurrence ci dessous :

Récurrence 1 (Needleman et Wunsch)

$$\mathcal{A}(S_1 = \overrightarrow{S_1}\alpha, S_2 = \overrightarrow{S_2}\beta) \quad = \quad max \begin{cases} \mathcal{A}(\overrightarrow{S_1}, S_2) + score(\alpha, '-') \\ \mathcal{A}(S_1, \overrightarrow{S_2}) + score('-', \beta) \\ \mathcal{A}(\overrightarrow{S_1}, \overrightarrow{S_2}) + score(\alpha, \beta) \end{cases}$$

(i) Algorithme d'Alignement Global de Needleman et Wunsch

Afin de résoudre le problème de l'alignement global optimal Needleman et Wunsch (Needleman and Wunsch, 1970) ont développé un algorithme permettant la construction d'une matrice de scores en temps quadratique en la taille des séquences en entrée. L'alignement global optimal entre deux séquences S_1 et S_2 de longueur $|S_1| = n$ et $|S_2| = m$ se décompose ici en deux étapes principales :

- Le calcul d'une matrice de score d'alignement M des deux séquences S_1 et S_2 de taille $(n+1) \times (m+1)$. Chaque coefficient de la matrice, $M[i, j]$ donne le score de l'alignement de $S_1[0, i-1]$ et $S_2[0, j-1]$. Ce score est calculé de manière itérative à partir des scores d'alignements des préfixes propres. Une fois la matrice M entièrement remplie, le coefficient $M[n, m]$ donne le score d'alignement global optimal des séquences S_1 et S_2.
- L'alignement global optimal est alors obtenu par reconstruction du chemin du calcul du score maximal (on parle de « backtrace »). Pour ce faire on part du coefficient $M[n, m]$ puis on remonte dans la matrice jusqu'à $M[0, 0]$.

Algorithme [*Algorithme de Needleman et Wunsch*] *Soient deux séquences S_1 et S_2 de longueurs respectives $|S_1| = n$ et $|S_2| = m$ et une fonction score. La recherche d'un alignement global optimal entre S_1 et S_2 suivant la fonction score est obtenu par construction d'une matrice des scores optimaux d'alignement $M_{n,m}$ telle que :*

- *Initialisation*

$$\begin{cases} M[0,0] = 0 \\ M[i,0] = M[i-1,0] + score(S_1[i-1], -), & \forall i \in [1,n] \\ M[0,j] = M[0,j-1] + score(-, S_2[j-1]), & \forall j \in [1,m] \end{cases}$$

- *Calcul du score optimal, pour $i \in [1,n]$ et $j \in [1,m]$*

$$M[i,j] = max \begin{cases} M[i-1,j-1] + score(S_1[i-1], S_2[j-1]) \\ M[i-1,j] + score(S_1[i-1], -) \\ M[i,j-1] + score(-, S_2[j-1]) \end{cases}$$

où M[i,j] représente le score de l'alignement de $S_1[0,i-1]$ avec $S_2[0,j-1]$ et M est une matrice de score. On obtient alors un alignement global optimal de S_1 avec S_2 en remontant de $M[n,m]$ à $M[0,0]$.

Si l'on considère la fonction de score d'identité unitaire s_u telle que $s_u(a,a) = 1$ et $s_u(a,b) = 0$ pour tout $a \neq b$ et qui pour chaque appariement correct de base affecte un score de 1 et pour chaque insertion ou délétion affecte un score de 0. L'alignement global de deux séquences revient à calculer la plus longue sous séquence commune entre les deux séquences.

On définit une sous-séquence comme suit :

Définition 8 (Sous-séquence)

Soit S une séquence de taille $|S|$ sur l'alphabet Σ et T une séquence de taille $|T|$ sur Σ telle que $|T| \leq |S|$. T est une sous-séquence de S s'il est possible d'obtenir T en supprimant $|S| - |T|$ symboles de S.

On note que les symboles de T ont le même ordre dans S et ne sont pas nécessairement consécutifs.

On peut alors introduire le problème de Longuest Common Subsequence (LCS) :

Définition 9 (Plus Longue Sous-Séquence Commune)

Soient deux séquences S_1 et S_2. $LCS(S_1, S_2)$ est la sous-séquence S_x de S_1 telle que :

- *S_x est aussi une sous-séquence de S_2.*
- *il n'existe pas de sous-séquence S_y de S_1 plus grande que S_x qui soit aussi sous-séquence de S_2.*

Comme nous l'avons remarqué, l'algorithme d'alignement global combiné à une fonction de score unitaire permet le calcul de la LCS en temps quadratique.

Le Problème d'Alignement Local Optimal

Dans de nombreux cas, on ne souhaite pas aligner deux séquences dans leur intégralité mais mettre en avant des sous parties de forte similarité. C'est dans cet objectif qu'a été introduit l'alignement local. L'alignement local a pour but de trouver des sous parties de S_1 et S_2 de plus forte similarité en accord avec la fonction de score. On parle de maximisation du facteur commun entre deux séquences.

L'alignement local est alors obtenu en comparant les facteurs calculés :

Définition 10 (Alignement Local)

Soient deux séquences S_1 et S_2, F_1 l'ensemble des facteurs de S_1, F_2 l'ensemble des facteurs de S_2 et la fonction $S(s_a, s_b)$ du score d'alignement global de s_a avec s_b. Le problème d'alignement local consiste à trouver $f_1 \in F_1$ et $f_2 \in F_2$ tels que $S(f_1, f_2)$ soit maximal.

De même que pour l'alignement global optimal, il existe une relation de récurrence entre l'alignement local des séquences S_1 et S_2 et l'alignement local de leurs plus longs préfixes propres, $\overrightarrow{S_1}$ et $\overrightarrow{S_2}$

(ii) Algorithme d'Alignement Local Optimal de Smith et Waterman

En 1981, Smith et Waterman ont développé un algorithme qui portera leurs noms et permettant le calcul de la matrice de score pour un alignement local (Smith and Waterman, 1981). La complexité de cet algorithme est $O(nm)$ où n et m sont les tailles des séquences dont on cherche l'alignement local maximal.

Tout comme pour l'alignement global optimal, l'alignement local optimal entre deux séquences S_1 et S_2 de longueur $|S_1| = n$ et $|S_2| = m$ se décompose en deux principales étapes :

- Le calcul de la matrice de score d'alignement M des deux séquences S_1 et S_2 est de taille $(n+1) \times (m+1)$. Chaque coefficient de la matrice, $M[i, j]$ donne le score de l'alignement local de $S_1[0, i-1]$ et $S_2[0, j-1]$. Ce score est calculé de manière itérative à partir des scores d'alignements locaux des préfixes propres.
- L'alignement local optimal est alors obtenu par reconstruction du chemin du calcul du score maximal (on parle de « backtrace »). Pour ce faire on part du coefficient $M[i, j]$ de valeur maximale puis on remonte dans la matrice jusqu'à $M[x, y] = 0$.

Algorithme [*Alignement local par programmation dynamique*] *Soient deux séquences S_1 et S_2 de longueurs respectives n et m. La recherche d'un alignement local optimal entre S_1 et S_2 suivant une fonction « score » est obtenu par construction d'une matrice des scores optimaux d'alignement $M_{n,m}$ telle que :*

- *Initialisation*

$$\begin{cases} M[0,0] = 0 \\ M[i,0] = 0, \quad \forall i \in [1,n] \\ M[0,j] = 0, \quad \forall j \in [1,m] \end{cases}$$

- *Calcul du score optimal, pour $i \in [1,n]$ et $j \in [1,m]$.*

$$M[i,j] = max \begin{cases} M[i-1,j-1] + score(S_1[i-1], S_2[j-1]) \\ M[i-1,j] + score(S_1[i-1], -) \\ M[i,j-1] + score(-, S_2[j-1]) \\ 0 \end{cases}$$

où M[i,j] représente le score de l'alignement de $S_1[0,i-1]$ avec $S_2[0,j-1]$ et M est une matrice de score. Pour obtenir le meilleur alignement local possible entre S_1 et S_2, on recherche la valeur maximale dans la matrice (possiblement multiple). Soit $M[k,l]$ une de ces valeurs. On remonte dans la matrice en suivant la relation de récurrence jusqu'à la première case $M[i,j]$ nulle. L'alignement local correspond alors aux facteurs $S_1[k,i]$ et $S_2[l,j]$.

Notons que si les régions alignées par l'algorithme de Smith-Waterman entre les deux séquences recouvrent l'intégralité des séquences, on peut alors considérer cet alignement local comme étant un alignement global.

Une version améliorée de l'algorithme de Smith-Waterman qui consiste à introduire une fonction de gap affine en fonction de la longueur du trou a été proposée par Osamu Gotoh. En effet, dans l'algorithme d'origine le coût d'un gap de longueur k est $k \times p$ (où p est la pénalité de gap), c'est donc un coût linéaire en fonction de la longueur du gap. Dans l'algorithme amélioré, ce coût se compose d'un coût d'ouverture de gap a et d'un coût d'extension de gap b, ainsi un gap de longueur k aura un coût de $a + (k-1)b$. Cette fonction de gap permet de favoriser le regroupement des gaps et donc d'éviter la multiplication des petits indels. Un raisonnement conforme à ce qu'il se passe au niveau biologique puisqu'il a été observé que les insertions et délétions sont majoritairement présentes aux niveaux des boucles à la surface de la structure de la protéine.

2.2 Chaînage dans les Séquences

L'un des principaux inconvénients des algorithmes d'alignement est leur complexité quadratique compliquant leur passage à l'échelle lors de l'analyse d'une séquence face à un grand ensemble de séquences. À cette fin, de nombreux « filtres » ont été introduits. Nous présenterons à la fin de ce chapitre les deux algorithmes les plus connus. Comme un grand nombre de ces filtres utilise le concept de chaînage, concept sur lequel repose également nos travaux, nous commencerons donc par détailler ce point.

2.2.1 Chaînage 1D en Séquence

Lorsqu'on dispose d'une séquence ADN et que l'on cherche par exemple à identifier les différents exons qui la composent le chaînage 1D est la méthode la plus appropriée (Gusfield, 1997). Dans cet exemple, on dispose d'un ensemble d'exons connus et de la séquence d'un gène à identifier. À l'aide de l'algorithme d'alignement, par exemple, on identifie des facteurs du gène similaires à certains exons. Un score est associé à chacun de ces facteurs. Pour finir, on va chercher un sous-ensemble de ces facteurs tel que les éléments de cet ensemble soient non chevauchants et que la somme de leurs scores soit maximale. L'identification de ce sous-ensemble correspond au problème de chaînage 1D dans une séquence.

Dans le cadre du chaînage 1D, on désignera par graine sur une séquence S, un facteur de S différent du mot vide.

Définition 11 (Chaînage 1D de séquences)

Soit S une séquence de taille $|S| = n$ et $G = g_1 \cdots g_m$ un ensemble de m graines sur S. À chaque graine g est associé un score. Calculer le meilleur chaînage consiste alors à trouver la combinaison de graines dont les occurrences sur S ne se chevauchent pas et dont la somme des scores est maximale.

L'algorithme permettant le calcul du chaînage 1D consiste à balayer les graines selon S. Lorsque l'on est sur le début d'une graine, alors on peut chaîner celle-ci avec la meilleure chaîne calculée jusqu'à présent. Lorsque l'on est à la fin d'une graine, on a établi la chaîne de meilleur score se terminant par cette graine. Si cette chaîne obtient un meilleur score que celui de la meilleure chaîne rencontrée jusqu'à présent, alors on met à jour celle-ci.

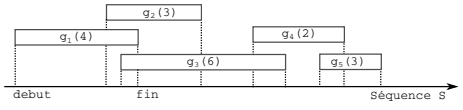


FIGURE 2.27 – *Exemple du problème de chaînage 1D avec cinq graines, g_1, g_2, g_3, g_4 et g_5, représentées ici par des rectangles. Les chaînes valides sont $[(g_1), (g_2), (g_3), (g_4), (g_5), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5)]$. La meilleure chaîne avec un score de 9 est (g_3, g_5).*

Ainsi on va disposer dans un tableau, P, l'ensemble des triplets $(pos, type, s)$, où s est l'identifiant de la graine, $type$ est soit $debut$ soit fin et pos la position de $type$ de g_s sur la séquence. On trie P selon les positions croissantes, pour des positions identiques, les positions de $type\ debut$ précèdent celles de $type\ fin$. On maintient également le score de la meilleure chaîne connue dans une variable, C. Afin de déterminer le meilleur chaînage on parcourt le tableau trié des positions :

- À chaque début de position d'une graine, celle-ci est chainée avec le meilleur chaînage déjà calculé. Le score associé à cette graine est mis à jour.
- À chaque position de fin d'une graine, on regarde si le chaînage calculé a un score supérieur à C. Si c'est le cas la valeur du meilleur chaînage, C, est mise à jour.

Cet algorithme présente une complexité en $O(m \log(m))$ ($O(m \log(m))$ pour le tri des positions, $O(m)$ pour le balayage).

2.2.2 Chaînage 2D en Séquences

Avec les avancées technologiques réalisées dans le séquençage, les NGS (« Next Generation Sequencing »), le nombre de séquences disponibles s'est rapidement accru ces dernières années. La nécessité d'analyser ces séquences et de comprendre leur implication, ou non, dans les fonctions cellulaires a joué un rôle moteur dans le développement de techniques de comparaison de séquences. Afin d'inférer les fonctions d'une séquence une méthode répandue consiste à comparer la nouvelle séquence identifiée à chacune des séquences d'une base de données de séquences connues. Dès lors de nombreux algorithmes de comparaison de séquences ont été développés. Le chaînage 2D est l'une de ces méthodes et consiste à comparer une séquence en entrée avec une seconde séquence. Le chaînage est par exemple une méthode pertinente de comparaison des génomes complets. En effet, aligner deux génomes entièrement n'est pas envisageable en raison de la longueur des séquences et de la divergence pouvant exister entre les deux génomes. Un algorithme de chaînage permet ainsi d'identifier en premier lieu les fragments de génomes communs et compatibles afin de pouvoir aligner les génomes en fonction de ces fragments. Au cours de cette partie nous poserons donc le problème de comparaison de séquences par chaînage de fragments, ou graines, communs. Nous focaliserons ensuite cette analyse sur deux des principaux algorithmes de chaînage 2D. Ceci donnera les bases de l'introduction du nouvel algorithme hybride que nous présenterons à la fin de cette section.

Définitions Préliminaires et Établissement du Problème

Comme pour le chaînage 1D, le chaînage 2D repose sur l'identification de graines, ici communes aux deux séquences S_1 et S_2 à comparer. De telles graines définissent un *hit*. Formellement, un hit h est défini par cinq éléments $(h.\ell, h.r, h.t, h.b, h.s)$: les quatre premières valeurs indiquent que les graines $S_1[h.\ell, h.r]$ (l et r, pour « left » et « right ») et $S_2[h.b, h.t]$ (t et b, pour « top » et « bottom ») forment un hit, le score associé à ce hit est la valeur $h.s$ (se référer à la Figure 2.28). On a alors $h.\ell \leq h.r$ et $h.b \leq h.t$. On appelle *extrémités* de h les coordonnées $(h.\ell, h.b)$ et $(h.r, h.t)$. Il est possible d'associer un hit à un rectangle dans le plan de comparaison, où l'axe des x correspond à S_1 et l'axe des y à S_2.

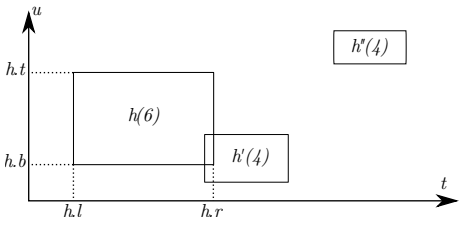


FIGURE 2.28 – *Exemple du problème de chaînage 2D avec trois hits, h, h′ et h″, représentés par des rectangles. Les chaînes possibles sont* $[(h), (h'), (h''), (h, h''), (h', h'')]$. *La meilleure chaîne avec un score de 7 est* (h, h'').

Soit maintenant $\mathcal{H} = \{h_0 \dots h_{m-1}\}$ un ensemble de hits, une *chaîne* est une liste ordonnée de hits $[h_1, \dots, h_\ell] \cup \mathcal{H}$ telle que $h_i.r < h_{i+1}.\ell$ et $h_i.t < h_{i+1}.b$ pour $i = 1, \dots, \ell$. Le score d'une telle chaîne est alors donné par la somme $\sum_{i=1}^{\ell} h_i.s$ de tous les hits qu'elle contient. D'un point de vue géométrique, tout comme les hits, une chaîne est naturellement incluse dans un rectangle contenant l'ensemble des hits de la chaîne.

On définit alors le problème du chaînage 2D comme suit :

Définition 12 (Chaînage 2D Optimal)

Soient S_1 et S_2 deux séquences de longueurs $|S_1|$ et $|S_2|$ et $\mathcal{H} = \{h_0 \dots h_{m-1}\}$ un ensemble de m hits sur S_1 et S_2. Trouver le chaînage 2D optimal revient à sélectionner le sous-ensemble $\mathcal{C}$ de $\mathcal{H}$ tel que :

- *$\mathcal{C}_1 = \{(l, r) / \exists t, b, s$ tels que $(l, r, t, b, s) \in \mathcal{C}\}$ est un chaînage 1D valide sur S_1 (idem pour S_2)*
- *pour tout couple $(h, h') \in \mathcal{C}$:*
  - *Si $h.l < h'.l$, alors $h.r < h'.l$.*
  - *Si $h.b < h'.b$, alors $h.t < h'.b$.*
- *Si $score(\mathcal{C}) = \sum_{h \in \mathcal{C}} h.s$, alors il n'existe pas de chaîne valide $\mathcal{C}' \cup \mathcal{H}$ telle que $score(\mathcal{C}') > score(\mathcal{C})$*

Nous allons maintenant présenter deux algorithmes permettant de résoudre le problème de chaînage 2D optimal. Les algorithmes exposés établissent le score de la chaîne optimale. Il est possible et facile de les modifier afin qu'ils calculent également la ou les chaînes optimales.

Algorithme de Programmation Dynamique du Chaînage 2D de Séquences

Nous présentons ici (voir le pseudo-code 1) un algorithme simple de programmation dynamique. Il permet le calcul, pour deux séquences S_1 et S_2, d'une matrice de programmation dynamique M de taille $|S_1| \times |S_2|$ telle que, pour tout $i \in [0, |S_1|-1]$ et $j \in [0, |S_2|-1]$, $M[i][j]$ est le score de la chaîne optimale incluse dans les préfixes $S_1[0, i]$ and $S_2[0, j]$.

Il existe une relation de récurrence entre la chaîne optimale sur $S_1[0, i]$ et $S_2[0, j]$ et les chaînes optimales de $S_1[0, k]$ et $S_2[0, l]$ pour $k \leq i$ et $l \leq j$. En effet, si aucun hit ne se termine en (i, j), alors $M[i][j] = max(M[i-1][j], M[i][j-1])$. Si le hit h est défini par $(h.l, h.r = i, h.b, h.t = j, h.s)$ alors la chaîne optimale se terminant en h a pour score $h.s + M[h.l-1][h.b-1]$.

Récurrence 2 (Chaînage 2D par Programmation Dynamique)

$$M[i][j] \quad = \quad max \begin{cases} M[i-1][j] \\ M[i][j-1] \\ \forall h \in \mathcal{H} \ t.q. \ h.r = i \ et \ h.t = j \\ \quad M[h.l-1][h.b-1] + h.s \end{cases}$$

En s'inspirant de la technique classique introduite dans (Hirschberg, 1975), nous présentons une version qui ne conserve pas l'intégralité de la matrice mais uniquement sa dernière colonne. Cela impose d'établir le score de la meilleure chaîne se terminant par le hit h non pas à la fin de ce hit, $(h.r, h.t)$ mais à son début, $(h.l, h.b)$.

Afin d'éviter un surcoût lors de la recherche des hits se terminant en (i, j), on utilise la matrice de listes chainées L.

Le point clé se situe dans la définition de $S[h]$ qui contient le score optimal de la chaîne qui contient h comme dernière graine.

L'initialisation de L se fait en $O(n_1 \times n_2)$. Le calcul de M se fait en $O(m+n_1 \times n_2)$, aussi l'algorithme présente une complexité totale en temps de $O(m+n_1 \times n_2)$ et une complexité en mémoire de $O(m + n_1 \times n_2)$. On observe donc que les performances de cet algorithme dépendent en grande partie de la taille des séquences en entrée. Nous allons maintenant voir un algorithme dont la complexité dépend uniquement du nombre de hits à chaîner.

Algorithme par Balayage du Chaînage 2D de Séquence

Nous nous intéressons maintenant à l'algorithme de chaînage par balayage. L'idée principale consiste à traiter les hits dans leur ordre d'apparition sur la séquence S_1, tout en maintenant à jour une structure de données conservant, pour chaque position j dans S_2, la meilleure chaîne partielle identifiée jusque là et composée uniquement des hits situés avant la position j.

Algorithme 1 L'Algorithme de Programmation Dynamique

```
1  L: une matrice de Listes chaînees de taille n_1 × n_2
2  S: un tableau de m entiers
3  M: un tableau de n_2 entiers
4  Pour tout h dans H Faire
5    Inserer en tete (h, fin) dans L[h.r][h.t]
6    Inserer en tete (h, debut) dans L[h.ℓ][h.b]
7  Pour i allant de 0 à n_1
8    gauche = 0
9    basGauche = 0
10   Pour j allant de 0 à n_2
11       maxC = 0
12       Pour tout (h, type) dans L[i][j]
13         Si type est debut
14           S[h] = h.s + basGauche
15         Si type est fin et S[h] > maxC
16           maxC = S[h]
17       basGauche = gauche
18       gauche = M[j]
19       M[j] = max(M[j], M[j − 1], maxC)
20 Retourner M[n_2 − 1]
```

Dans cet algorithme, P conserve toutes les valeurs des extrémités de l'ensemble des hits et $S[s]$ est le score de la chaîne optimale parmi toutes les chaînes qui se terminent par le hit h, tout comme dans l'algorithme de programmation dynamique. Un hit est dit traité lorsque l'entrée $(h.r, end, h.s)$ a elle-même été traitée par la boucle à la ligne 8. Une *chaîne partielle* est une chaîne uniquement constituée de hits traités.

Afin d'assurer l'exactitude de l'algorithme, la structure de données nommée A doit satisfaire les besoins invariants suivants : si $(pos, type, s)$ est la dernière entrée de P qui a été traitée, alors A contient une entrée (p, v) si et seulement si la meilleure chaîne, parmi toutes les chaînes partielles qui appartiennent au rectangle défini par les positions $(0, 0)$ et (pos, p), est v et correspond à une chaîne se terminant par le hit h' tel que $h'.t = p$. Ceci est assuré par la ligne 16. Cet invariant permet de retrouver à partir de A (voir ligne 11) le score d'une chaîne optimale partielle qui peut être étendue par le hit courant h (c'est-à-dire qui se termine sur la séquence S_2 avec une position strictement inférieure à $h.b$ puisque l'ordre dans lequel les graines sont traitées assure que tous les hits précédemment traités ne chevauchent pas le hit en cours sur la séquence S_1.

Du point de vue de la complexité, il est fondamental de s'assurer que sur la ligne 16, le temps requis pour supprimer les c entrées (l'ensemble des entrées de A ayant une première valeur strictement supérieure à p et inférieure ou égale à p') est en

Algorithme 2 Comparaison 2D en séquence : balayage

L'Algorithme par Balayage

```
1  P: un tableau de 2m triplets (position, type, graine)
2  A: un ensemble de paires (position, score)
3  S: une tableau de m entiers
4  Pour tout h dans H Faire
5     Inserer (h.ℓ, debut, h.s) dans P
6     Inserer (h.r, fin, h.s) dans P
7  Trier P selon le champs position, avec les positions de debut avant lespositions de fin
       pour des valeurs identitiques
8  Pour tout (pos, type, s) dans P
9     Si type est debut
10       Extraire de A la paire (p, v) telle que p est la plus grande position strictement
             inférieure à h.b
11       S[h] = h.s + v
12    Si type est fin
13       (p, v) = Extraire de A la plus grande position inférieure ou égale à h.t
14       Si S[h] > v
15          Extraire de A la paire (p', v') telle que v' est le score le plus eleve inferieure ou
                égale à S[h]
16          Retirer de A toutes les entrees (p'', v'') telles que p < p'' ≤ p'
17          Inserer (h.t, S[h]) dans A
18 (p, v) = derniere entree de A
19 Retourner v
```

$O(c\log(m))$. Si la structure de données implémentée pour A satisfait cette propriété et supporte les recherches, les insertions, les délétions, en un temps logarithmique, alors la complexité en temps de cet algorithme est $O(m\log(m))$ ((Höhl et al., 2002) discute de telles structures de données). Parmi les structures possibles, on pourra utiliser un arbre de type AVL. À noter, le tri initial de P prend également un temps en $O(m\log(m))$. Du point de vue mémoire, la complexité de l'algorithme est linéaire.

On observe ici que la complexité totale de l'algorithme par balayage dépend uniquement du nombre de hits à traiter et est indépendante de la taille des séquences en entrée.

2.2.3 Algorithme Hybride de Chaînage 2D en Séquences

Nous allons maintenant nous intéresser à la combinaison de ces deux principes afin de bénéficier des avantages de chacun d'eux. En effet, théoriquement le nombre de hits m peut être quartique et il peut ainsi arriver que $O(m\log m)$ soit supérieur à $O(m + n_1 \times n_2)$. Une approche naïve consisterait alors à comparer $m + n_1 \times n_2$ et $m\log m$ afin de décider quel algorithme employer. Outre le fait qu'une telle comparaison ne permettrait pas d'obtenir de façon certaine la meilleure performance (du fait des constantes dans les complexités), il peut arriver que la *densité* des hits varie en fonction de la position sur la séquence. Ce qui suggère que, dans les régions présentant une forte densité de hits, il serait plus efficace d'employer l'algorithme de programmation dynamique, alors que dans les régions de faible densité en hits, l'algorithme par balayage serait plus efficient. De plus, dans le cas de l'algorithme par balayage, les hits sont traités dans leur ordre d'apparition sur la séquence S_1, ce qui est également l'ordre respecté par l'algorithme dynamique pour remplir les colonnes de la matrice de programmation dynamique. Ainsi, les hits sont analysés dans le même ordre par les deux méthodes. Ce qui soulève la question à laquelle nous allons répondre dans cette section ; peut-on calculer une chaîne optimale en traitant les hits dans l'ordre de l'algorithme dynamique et de l'algorithme par balayage en alternant entre l'équation classique de programmation dynamique et l'approche par balayage en fonction de la densité de hits à la position courante sur S_1 ?

Avant d'établir les principaux résultats de cette section, nous introduisons d'abord la notion d'*instance compacte*.

Une instance du problème de chaînage est dite *compacte* si chaque position de S_1 ou S_2 contient au moins une extrémité de hit. Si une instance n'est pas compacte, alors il existe une unique instance compacte obtenue en supprimant de S_1 et S_2 toutes les positions qui ne contiennent pas une extrémité de hit. Cette instance compacte mène aux séquences S'_1 et S'_2 de tailles respectives $|S'_1| = n'_1$ et $|S'_2| = n'_2$. La mise à jour des extrémités des hits selon les séquences S'_1 et S'_2 genère un jeu $\mathcal{H}'$ de hits. Dès lors, on dénote par $(S'_1, S'_2, \mathcal{H}')$ une instance compacte de $(S_1, S_2, \mathcal{H})$.

Dans un deuxième temps, on définit pour une position p de S_1 la densité d'extrémité $\mathcal{K}_p$ comme le nombre d'extrémités de hits en cette position. Si on appelle P^1 l'ensemble des positions de S'_1 dont le nombre d'extrémités est strictement supérieur

à $\frac{n'_2}{\log n'_2 - 1}$ et P^2 les $n'_1 - |P^1|$ positions de S'_1 restantes, alors la complexité en temps de l'algorithme hybride que nous allons présenter est

$$O\left(m + \min(m \log m, n_1) + \min(m \log m, n_2) + \sum_{p \in P^1} (n'_2 + \mathcal{K}_p) + \log n'_2 \sum_{p \in P^2} \mathcal{K}_p\right).$$

De manière intuitive notre algorithme hybride prend pour point d'appui l'instance compacte du problème, et complète la matrice de programmation dynamique pour cette instance compacte en décidant pour chaque colonne de la matrice (c'est-à-dire pour chaque position de S'_1) de la remplir à l'aide de l'équation de programmation dynamique ou au contraire avec l'algorithme glissant, en se basant sur la densité d'extrémités.

Problème de Compaction

La première amélioration envisagée consiste à considérer l'instance compacte $(S'_1, S'_2, \mathcal{H}')$.

Lemme L'instance compacte $(S'_1, S'_2, \mathcal{H}')$ peut être calculée en $O(m + \min(m \log m, n_2) + \min(m \log m, n_1))$ en temps et $O(m + n_1 + n_2)$ en espace.

La preuve de ce lemme est simple, et nous n'entrerons pas dans les détails. Le principe consiste à traiter S_1 puis S_2 pour éliminer, et ce dans chaque séquence, les positions qui ne contiennent aucun bord de hit. Supposons que nous traitions S_1. Si $m \log m \leq n_1$, alors il est seulement nécessaire de classer les extrémités des hits, de regrouper les extrémités possédant la même valeur et de renommer chaque extrémité par le nombre de groupes qui la précède, lorsqu'ils sont classés, plus un. Si $m \log m > n_1$, alors dans ce cas il est seulement nécessaire de commencer par détecter les positions de S_1 qui ne sont pas des extrémités de hit, en $O(m + n_1)$ en temps, de marquer ces positions puis de renommer les positions dont la densité est supérieure à zéro, en $O(n_1)$ en temps. Enfin, pour terminer, on renomme les extrémités des hits en fonction des nouvelles étiquettes des positions sur S_1, en $O(m)$ en temps.

Ainsi, l'utilisation de l'algorithme de programmation dynamique sur l'instance compacte présente une complexité de $O(m + \min(m \log m, n_1) + \min(m \log m, n_2) + n'_1 \times n'_2)$ en temps et $O(m + n_1 + n_2)$ en espace.

À partir de maintenant, on suppose que l'instance compacte a été calculée et qu'elle constitue l'instance considérée.

Choix de l'Algorithme : par Balayage ou par Programmation Dynamique

Nous abordons ici l'idée principale de cet algorithme hybride. Le principe repose sur le traitement des hits dans le même ordre que dans l'algorithme par balayage-soit en utilisant une boucle qui itère de l'indice 0 à l'indice $n'_1 - 1$, une caractéristique commune aux deux algorithmes, par balayage et de programmation dynamique.

Pour traiter un hit, dont une extrémité sur S'_1 est à la position i, on utilise : soit les équations de programmation dynamique, soit le principe d'algorithme par balayage. La décision concernant l'approche à adopter doit être prise en fonction de la densité (c'est-à-dire du nombre d'extrémités de hits) à la position i de S'_1 qui correspond à la colonne analysée. D'où les conditions nécessaires suivantes :

- Lorsque l'on utilise l'approche par programmation dynamique, il est possible d'avoir accès à la colonne précédente de la table de programmation dynamique.
- Lorsque l'on utilise l'approche par balayage, il est possible d'avoir accès à une structure de données permettant de répondre aux mêmes requêtes qu'avec la structure de données A de l'algorithme par balayage.

Afin de respecter ces conditions, nous introduisons une nouvelle structure de données.

Une structure de données hybride Pour permettre de faire appel à chacune des deux approches mises en oeuvre, nous introduisons une structure de données B qui est principalement un tableau de n'_2 entrées complété par un arbre binaire équilibré. De manière plus formelle, considérons un tableau $\mathcal{B}$ de n'_2 entrées, tel que $\mathcal{B}[i]$ contient un score de chaînage et satisfait la condition suivante : Si h est le dernier hit traité, pour tout $i = 1, \ldots, h.r$, $\mathcal{B}[i] \geq \mathcal{B}[i-1]$. Ce tableau est complété par une structure d'arbre binaire équilibré dont les feuilles sont les entrées de $\mathcal{B}$ et les noeuds internes sont étiquetés de manière à satisfaire l'invariant suivant : un noeud x a pour étiquette la valeur de l'étiquette maximale prise parmi l'ensemble de ses deux fils droit et gauche.

On note B cet arbre binaire. Cette structure de données est utilisée de manière similaire à la structure de données A de l'algorithme par balayage ; il permet de répondre aux questions suivantes : soit $0 \leq p \leq n'_2$, on cherche le score optimal d'une chaîne partielle dont le dernier hit h vérifie $h.t \leq p$.

Nous allons maintenant décrire comment implémenter cette structure de données dans un tableau. Soit b le plus petit entier tel que $n'_2 \leq 2^b$. On encode alors B dans un tableau de taille 2^{b+1}, dont le préfixe de taille $2^b - 1$ contient les étiquettes des noeuds internes de l'arbre binaire, ordonnées d'abord selon un parcours en largeur, alors que les entrées de $\mathcal{B}$ (voir la Figure 2.29) sont conservées dans le suffixe de taille n'_2 du tableau. Ainsi $B[0]$ correspond à la racine de l'arbre binaire, $B[1]$ est le fils gauche de cette racine et $B[2]$ est son fils droit. Également $B[2^b]$, qui correspond à $\mathcal{B}[0]$, est le fils le plus à gauche de l'arbre et $B[2^b + n'_2 - 1]$, qui correspond à $\mathcal{B}[n'_2 - 1]$, est le fils le plus à droite de l'arbre.

Cet encodage nous permet d'obtenir facilement et en temps constant le fils gauche, le fils droit, le parent et le fils le plus à droite d'un noeud interne x. Dans un tel cas, le fils le plus à droite de x est la feuille atteinte après avoir suivi le chemin issu de x et ne passant que par les arêtes menant à des fils droits. En effet, pour le noeud à la position x du tableau, si $x >= 2^b - 1$, x est une feuille. Sinon, soit y le plus grand entier tel que $2^y \leq x + 1$ et soit $z = x - 2^y + 1$:

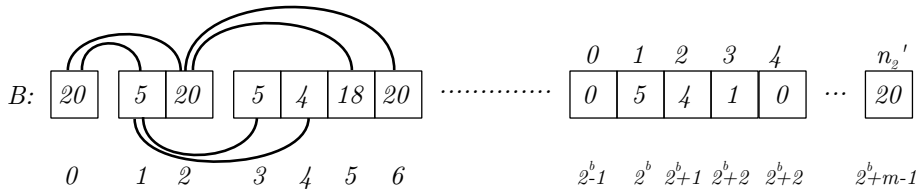


Figure 2.29 – *Encodage de l'arbre binaire B dans un tableau.*

- $filsGauche(x)$: le fils gauche de x est $2^{y+1} - 1 + 2 * z$
- $filsDroit(x)$: le fils droit de x est $2^{y+1} - 1 + 2 * z + 1$
- $parent(x)$: si x vaut zéro, x est la racine (retourne -1), sinon son parent est $2^{y-1} - 1 + |\frac{z}{2}|$
- $filsLePlusADroite(x)$: le fils le plus à droite de x est $2^b - 1 + 2^{b-z} * (z+1) - 1$

L'algorithme par balayage et la structure de données hybride Pour mettre à jour la structure de données, lorsqu'un hit h a été traité selon l'approche par balayage, nous utilisons la fonction $fixerScore$ (voir l'Algorithme 3), avec les paramètres $p = h.t$ et $score = S[h]$. Il est alors évident que, si toutes les modifications de B sont réalisées grâce à la fonction $fixerScore$, alors les deux conditions nécessaires à B sont satisfaites. De plus, il est également clair que la recherche du meilleur score de chaîne partielle se terminant sur S'_2, à une position strictement inférieure à p, peut être résolue par la fonction $meilleursScore$ décrite dans l'Algorithme 4. Cette fonction constitue une simple recherche dans un arbre binaire.

Algorithme 3 Calcul du score de chaînage à la position p.

```
1 fixerScore(B, p, score) :
2   index = 2^b − 1 + p // on debute à partir de la feuille correspondant à p
3   Tant que index! = −1 && B[index] < score
4         B[index] = score
5         index = parent(index)
```

La complexité en temps des deux algorithmes $fixerScore$ et $meilleurScore$ est en $O(\log(n'_2))$ puisque l'arbre binaire est équilibré.

Maintenant, il est possible d'implémenter l'algorithme par balayage sur une instance compacte en utilisant la structure de données B. Pour cela, dans l'algorithme par balayage de chaînage 2D, il suffit :

- de remplacer les instructions à la ligne 11 par un appel à la fonction $meilleurScore(B, s.b)$.
- de remplacer le bloc d'instructions aux lignes 13-17 par la fonction $fixerScore(B, S[s])$
- de lire le score de la chaîne optimale sur l'étiquette de la racine de l'arbre binaire

Algorithme 4 Recherche du meilleur score de chaînage pour une chaîne partielle terminant strictement sous la position p.

```
1  meilleursScore(B, p) :
2    Soit b le plus petit entier t.q. n'_2 ≤ 2^b
3    maxScore = 0
4    noeudCourant = 0 // le noeud racine
5    indexOfP = 2^b − 1 + p
6    Tant que filsLePlusADroite(noeudCourant) > indexOfP
7          gauche = filsGauche(noeudCourant)
8          Si filsLePlusADroite(gauche) >= indexOfP// deplace gauche
9            noeudCourant = gauche
10         Sinon // deplace droite
11           maxScore = max(maxScore, B[gauche])
12           noeudCourant = filsDroit(noeudCourant)
13   retourner max(maxScore, B[noeudCourant])
```

La complexité des opérations réalisées sur B est logarithmique et dépend de n'_2, qui est inférieur ou égal à m. La complexité finale est donc de $O(m \log n'_2)$.

L'algorithme par programmation dynamique et la structure de données hybride Inversement, nous pouvons implémenter l'algorithme de programmation dynamique avec la structure de données B en utilisant $\mathcal{B}$ comme la colonne courante de la matrice de programmation dynamique -*i.e.* si la position de S'_1 en cours de traitement est i, $\mathcal{B}[j]$ est le score de la meilleure chaîne partielle inclue dans le restangle défini par $(0, 0)$ et (i, j)- sans mettre à jour les noeuds internes de l'arbre binaire.

L'approche Balayage/Dynamique avec la structure de données hybride Ainsi, soit un algorithme hybride qui repose sur la structure de données B, lorsque l'on change d'approche algorithmique (de l'algorithme par balayage au dynamique ou inversement), la structure de données B doit demeurer cohérente pour l'approche en cours d'application puis elle doit être mise à jour pour devenir cohérente pour l'approche suivante.

Ainsi lorsque l'on passe de l'algorithme de programmation dynamique à l'algorithme par balayage (soit lorsque l'on utilise l'approche par balayage à la position i alors que l'approche par programmation dynamique a été appliquée pour la position $i-1$), $\mathcal{B}[j]$ $(j = 0, \ldots, n'_2 - 1)$ est le score optimal de la chaîne partielle du rectangle défini par $(0, 0)$ et $(i - 1, j)$. Nous désirons également mettre à jour B de manière à ce que la valeur de l'étiquette de tout noeud interne x de l'arbre binaire soit la valeur maximale prise parmi ses deux fils. Comme $\mathcal{B}$ représente les feuilles de l'arbre binaire, cette mise à jour peut être opérée au cours d'un parcours post-fixe de l'arbre binaire, soit avec une complexité de $O(n'_2)$.

Lorsque l'on passe de l'algorithme par balayage à l'algorithme de programmation dynamique (soit lorsque l'on utilise l'approche de programmation dynamique à la position i alors que l'approche par balayage a été appliquée pour la position $i-1$), pour toute chaîne optimale partielle incluse dans $(0,0)$ et $(i-1, n'_2-1)$ et dont le dernier hit est h, nous avons le score de cette chaîne dans la feuille $\mathcal{B}[h.t]$. Cela provient directement de la manière dont les étiquettes sont ajoutées dans l'arbre binaire par la fonction $fixerScore$. Pour mettre à jour B, nous avons besoin en réalité que $\mathcal{B}[j]$ contienne le score optimal d'une chaîne partielle dont le dernier hit se termine *au plus* à la position j. Dans cette optique, la fonction de mise à jour nécessite seulement d'affecter à $\mathcal{B}[j]$ la valeur $\max(\mathcal{B}[j-1], \mathcal{B}[j])$ pour j allant de 1 à n'_2-1. Cette opération se fait en $O(n'_2)$.

Ainsi, mettre à jour la structure de données B de l'algorithme par balayage à l'algorithme dynamique ou inversement, de l'algorithme dynamique à l'algorithme par balayage, peut être fait en $O(n'_2)$. On appelle *actualiser* la fonction permettant cette mise à jour.

Choix de l'algorithme par balayage ou dynamique selon la densité en hits Avant d'introduire notre algorithme hybride, il est nécessaire d'expliciter un point clef important : Comment choisir le modèle (algorithme par balayage ou algorithme dynamique) à employer lorsque l'on traite les hits présentant une extrémité à la position étudiée sur S_1, appelée i. Soit $\mathcal{K}_i$, le nombre de hits h telles que $h.\ell = i$ ou $h.r = i$. Lorsque l'on utilise l'approche dynamique, le coût de la mise à jour de $\mathcal{B}$ (soit de calculer la colonne i de la matrice de programmation dynamique) est en $O(n'_2 + \mathcal{K}_i)$. Avec l'algorithme par balayage, le coût de la mise à jour de B est en $O(\mathcal{K}_i \log n'_2)$.

D'où, si $\mathcal{K}_i > \frac{n'_2}{\log n'_2 - 1}$, le coût asymptotique de l'approche dynamique est meilleur que le coût asymptotique de l'approche par balayage ; Alors que l'on observe le phénomène inverse si $\mathcal{K}_i \leq \frac{n'_2}{\log n'_2 - 1}$. Par conséquent, on conserve dans un tableau C de taille n'_1 l'information `DP` ou `LS` selon que la position correspondante sur S'_1 doit être traitée suivant l'approche par programmation dynamique ou par balayage. Le calcul de C se fait en $O(m)$.

Cette dernière observation conduit à l'algorithme hybride final (voir l'Algorithme 5).

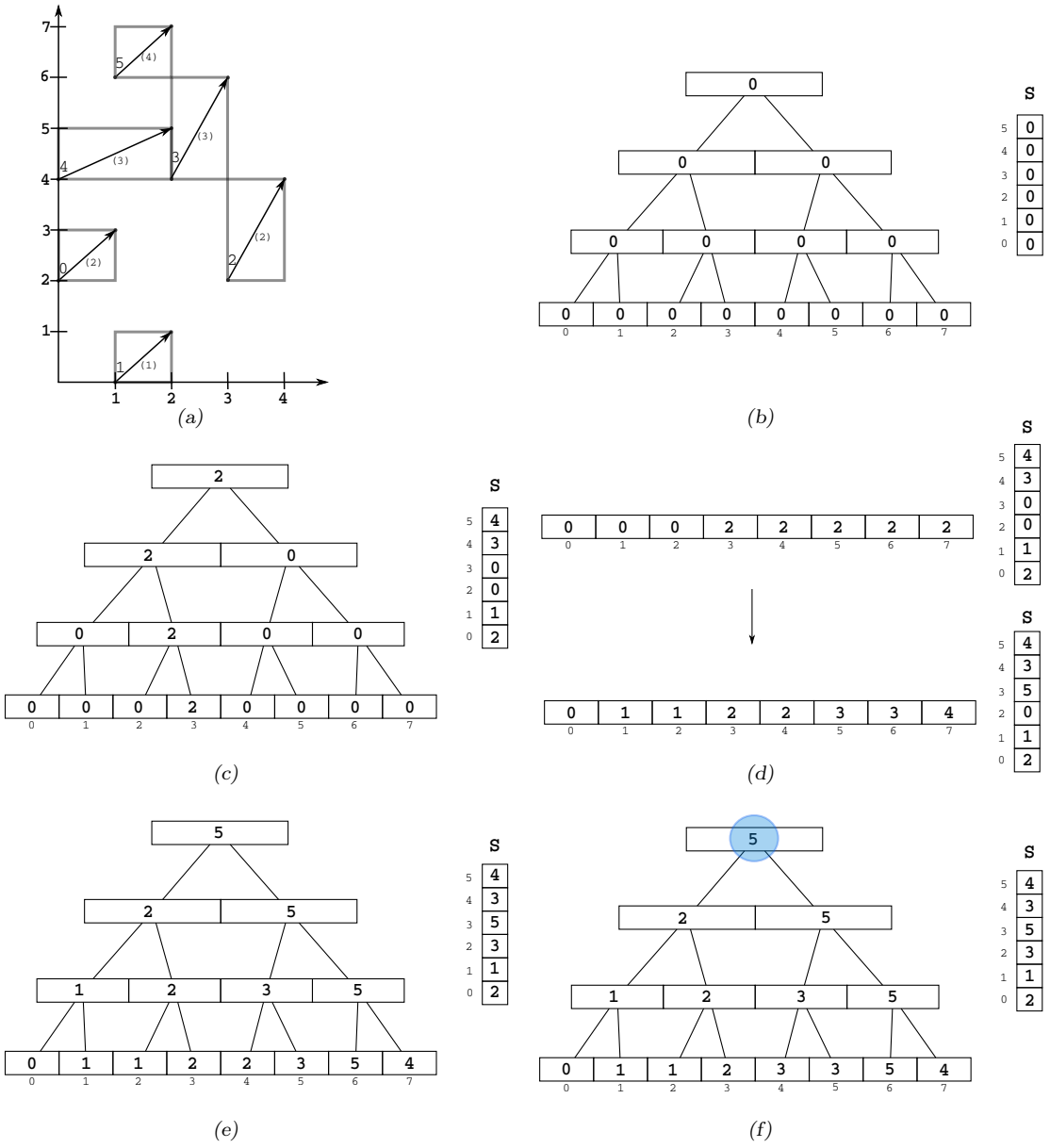


FIGURE 2.30 – *Application de l'algorithme hybride à un exemple de six graines. (a) Représentation des graphique de cinq graines :* $\{(0,2), (1,1), (2,2), (3,3), (4,3), (5,4)\}$. *(b) Initialisation de l'arbre binaire et du tableau des feuilles de cet arbre. (c) Arbre binaire après le traitement par balayage de la position 1. (d) Transition traitement par balayage traitement dynamique et état du tableau des feuilles après le traitement de la position 2. (e) Arbre binaire après traitement de la position 3. (f) Arbre final après traitement de la position 4. La racine de l'arbre donne le score maximal de chaînage. Chacune des étapes est accompagnée du tableau de scores.*

Algorithme 5 Un algorithme hybride pour le problème de chaînage 2D de hits.

```
1  //Calcul de l'instance compacte (S'_1, S'_2, H')
2  L: un tableau de n'_1 × n'_2 de listes chainées
3  C: un tableau binaire de taille n'_1
4  //Calcul de C
5  Pour tout h dans H' faire
6    Si C[h.r] est DP → inserer devant (h, fin) dans L[h.r][h.t]
7    Sinon → inserer devant (h, fin) dans L[h.r][0]
8    Si C[h.ℓ] est DP → inserer devant (h, debut) dans L[h.ℓ][h.b]
9    Sinon → inserer devant (h, debut) dans L[h.ℓ][0]
10
11 B: un arbre binaire de n'_2 feuilles (tous les noeuds sont initialisés avec la valeur zero)
12 S:un tableau d'entiers de taille m
13 B: fait référence aux n'_2 feuilles de B
14 Pour i allant de 0 a n'_1 − 1
15   Si C[i] ≠ C[i − 1]
16     actualiser(B)
17   Si C[i] est DP
18     gauche = 0, basGauche = 0
19     Pour j allant de 0 à n'_2 − 1
20       maxC = 0
21       Pour tout (h, type) dans L[i][j]
22         Si type est debut
23           S[h] = h.s + basGauche
24         Si type est fin et S[h] > maxC
25           maxC = S[h]
26         basGauche = gauche, gauche = B[j]
27         B[j] = max(B[j], B[j − 1], maxC)
28 Sinon // C[i] est LS
29   Pour tout (h, type) dans L[i][0]
30     Si type est debut
31       S[h] = h.s + meilleursScore(B, h.b)
32     Si type est fin
33       fixerScore(B, h.t, S[h])
34 Si C[n'_1 − 1] est DP → retourner B[n'_2 − 1]
35 else → retourner la valeur de la racine de B
```

Complexité en Temps et en Espace

En terme de complexité en espace, du faite de la matrice de listes chainées L, l'Algorithme 5 est en $O(m + n'_1 \times n'_2)$. Cependant, de même que nous calculons la matrice de score sur une seule colonne, il est possible d'utiliser une seule liste chainée mise à jour au début de la boucle principale (ligne 14). Ainsi, la complexité en espace de l'algorithme n'est plus que de $O(m + n'_1 + n'_2)$.

Nous nous intéressons maintenant à la complexité en temps de cet algorithme hybride. Si la position i de S'_1 est étiquetée comme `DP`, le coût de mise à jour de la colonne est $O(n'_2 + \mathcal{K}_i)$. Si $C[i]$ est `LS`, le coût du calcul des scores des chaînes à cette position est $O(\mathcal{K}_i \log n'_2)$ (lignes 28–33). Donc, si nous appelons P^1 l'ensemble des positions sur S_1 pour lesquelles nous utilisons l'approche dynamique (`DP`), P^2 l'ensemble des positions de S'_1 pour lesquelles nous employons l'algorithme par balayage (`LS`), le temps nécessaire à l'exécution de toute la boucle à la ligne 14 est

$$O\left(\sum_{p\in P^1}(n'_2 + \mathcal{K}_p) + \sum_{p\in P^2}\mathcal{K}_p \log n'_2\right)$$

Nous avons alors $|P^1| + |P^2| = n'_1$, $\forall p \in P^1 : \mathcal{K}_p > \frac{n'_2}{\log n'_2 - 1}$ et $\forall p \in P^2 : \mathcal{K}_p \leq \frac{n'_2}{\log n'_2 - 1}$. De plus, la mise à jour de la structure de données B lors du passage de l'approche progressive à l'approche dynamique ou inversement (ligne 16) est réalisée au plus une fois de plus que la taille de P^1, en conséquence le coût total de cette opération est $O\left(\sum_{p\in P^1} n'_2\right)$, et peut donc être intégré, asymptotiquement, au coût des traitements des positions de P^1.

Théorème 1 [`Algorithme hybride`]

L'algorithme hybride complet (compaction incluse) calcule le score d'une chaîne optimale avec une complexité en temps :

$$O\left(m + \min(m\log m, n_1) + \min(m\log m, n_2) + \sum_{p\in P^1}(n'_2 + \mathcal{K}_p) + \log n'_2 \sum_{p\in P^2}\mathcal{K}_p\right) \tag{2.1}$$

et une complexité en espace :

$$O(m + n_1 + n_2).$$

Pour conclure cette analyse de complexité, nous montrons que l'algorithme hybride est au moins aussi performant que les deux algorithmes dont il s'inspire, à savoir l'algorithme par balayage et l'algorithme de programmation dynamique.

De l'Équation (2.1), nous pouvons déduire que, si $P^1 = \emptyset$, la complexité en temps de l'algorithme devient

$$O(m + \min(m \log m, n_1) + \min(m \log m, n_2) + \log n'_2 m)$$

Ce qui, dans le pire des cas, comme $n'_2 = \min(n_2, m)$, revient à la complexité en temps de l'algorithme par balayage.

Maintenant, si $P^1 \neq \emptyset$, pour chaque position i de P^1, nous savons que le coût de mise à jour de B et du traitement de i par l'algorithme de programmation dynamique n'est pas pire que par le traitement avec l'algorithme par balayage pour une valeur $\mathcal{K}_i$ choisie. Cela garantit une performance de l'algorithme hybride au moins égale aux performances de l'algorithme par balayage.

On considère maintenant l'algorithme de programmation dynamique. De nouveau, à partir de l'Équation (2.1), si $P^2 = \emptyset$, la complexité devient

$$O(m + \min(m \log m, n_1) + \min(m \log m, n_2) + n'_1 \times n'_2)$$

Ce qui est au plus égal à la complexité de l'algorithme de programmation dynamique original puisque $n'_1 = \min(n_1, m)$ et $n'_2 = \min(n_2, m)$.

Comme ci-dessus, si on suppose maintenant que $P^2 \neq \emptyset$, alors nous savons que le coût de traitement des positions de P^2 par l'algorithme par balayage n'est asymptotiquement pas pire que leur traitement par l'algorithme de programmation dynamique. Le coût de mise à jour de B lors de la permutation de l'algorithme dynamique avec l'algorithme par balayage peut être intégré au coût asymptotique de la partie avec l'algorithme dynamique. Cela souligne que l'algorithme hybride n'est, asymptotiquement, pas pire que l'algorithme de programmation dynamique pure.

2.3 Deux Principaux Filtres d'Alignement

Nous venons de voir l'approche par chaînage qui permet rapidement de comparer deux séquences une fois un ensemble de hits établi. Le résultat du chaînage peut servir de base à un algorithme de type alignement contraint par les hits constituant la chaîne optimale, ce qui permet d'affiner la comparaison. Aussi un élément important de cette approche est la recherche des hits. L'ensemble constitue ce que l'on appelera un *filtre*. Les travaux que nous développerons dans le chapitre 3 reposent sur ce schéma méthodologique.

Pour clore ce chapitre nous allons voir deux des filtres les plus utilisés pour l'analyse de séquences, à savoir l'algorithme *FastA* et l'algorithme *BLAST*.

2.3.1 Algorithme FastA

A l'origine David J. Lipman et William R. Pearson développèrent en 1985 (Lipman and Pearson, 1985) un programme de recherche de similarités entre deux séquences protéiques (protéine *vs* protéine) noté *FAST-P*. La possibilité de comparer

des séquences nucléiques (ADN *vs* ADN), version *FAST-N* du programme, mais également la possibilité de rechercher des similarités entre séquences nucléiques et protéiques (ADN traduit *vs* protéine) ont alors été ajoutées lors d'améliorations (Pearson and Lipman, 1988). Le programme *FastA* pour FAST-All est alors une extension des programmes précédents et fonctionne sur les deux alphabets protéique et nucléique. Dans cette amélioration du programme une nouvelle méthode de calcul du score est également apportée. Actuellement FastA est une suite de programmes d'alignements (*prot/prot*, *ADN/ADN*, *prot/ADN traduit* avec prise en charge des six phases de lecture) et inclut des algorithmes de recherche par traduction et le programme SEARCH qui est une implémentation de l'algorithme de Smith-Waterman. Cette suite de programmes a pour objectif de mettre à disposition des méthodes de calcul de statistiques de similarité pour juger de la pertinence des alignements. Il est intéressant de noter que le format de fichier FastA utilisé comme fichier d'entrée pour la suite FastA est devenu par sa très large utilisation un format standard. Ce format est également utilisé par d'autres programmes de comparaison, d'alignements de séquences et de recherche de séquences dans des bases de données.

FastA utilise des alignements locaux afin d'identifier, dans une base de données, un ensemble de séquences cibles similaires à la séquence en entrée. Pour cela il suit une heuristique de recherche des correspondances de groupes d'acides aminés (respectivement nucléotides) consécutifs de longueur k ou k-mer.

Définition 13 (k-mer)

Soit une séquence S de longueur $|S| = n$, un k-mer *est une séquence de longueur k sur S où $k \leq n$. Un k-mers est donc un facteur de longueur k. Dans la séquence S il y a donc $n - k + 1$ k-mer (dont certains peuvent être identiques).*

Dans un premier temps, l'ensemble des k-mers communs entre la séquence en entrée et chacune des séquences de la base de données, ou séquences cibles, est identifié. Après une suite de traitements sur ces k-mers communs que nous allons détailler par la suite, il résulte un ensemble de contraintes entre l'entrée et certaines des cibles. Ces contraintes sont utilisées dans une variante de l'algorithme de Smith-Waterman pour calculer les alignements finaux. La longueur k permet de contrôler la sensibilité et la rapidité de la comparaison : plus k est grand plus le nombre de résultats décroît mais plus la rapidité du programme augmente. Le choix de la valeur de k est alors un compromis entre vitesse et sensibilité.

Le filtre FastA se découpe en cinq étapes, détaillées ci-après :

1 Recherche des k-mers entre deux séquences.
2 Identification des régions de diagonale.
3 Association de scores aux meilleurs régions à l'aide d'une matrice de substitution. On note les meilleures « régions initiales » `INIT1`.
4 Chaînage de plusieurs « régions initiales ». On obtient `INITN`.
5 Alignement final à l'aide d'un algorithme proche de Smith-Waterman.

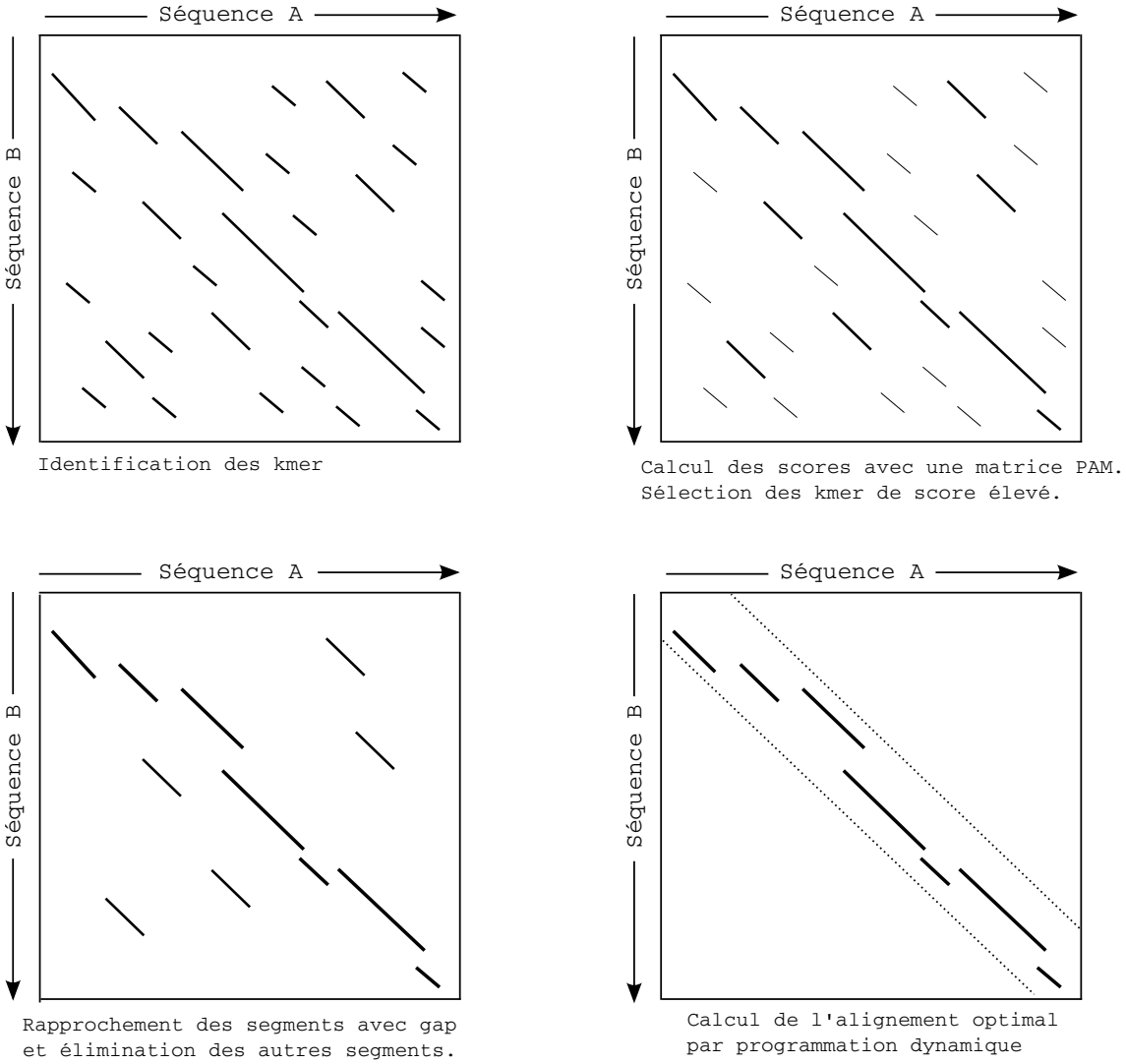


Figure 2.31 – *Principe de l'algorithme FastA.*

(i) Recherche des k-mers

L'idée de cette première étape provient du fait que, dans la plupart des cas, les séquences homologues contiennent au moins quelques segments, plus ou moins longs, identiques.

Dans un premier temps, l'utilisateur choisit une valeur pour le paramètre k. Au cours de la première étape de l'heuristique, on identifie les motifs de longueurs k comme suit. FastA identifie les paires (i, j) telles que le motif de longueur k, que nous appellerons ici k-mer, débutant à la position i sur la séquence requête S_1 est parfaitement identique au k-mer débutant à la position j sur la séquence cible S_2. Une telle paire correspond à un hit (appelée « hot spot » dans la terminologie FastA).

Par défaut FastA utilise des k-mers de taille 2 pour les acides aminés et de taille

6 pour les acides nucléiques. Des valeurs plus élevées de k-mers accroissent la vitesse (moins de k-mers sont pris en considération dans les étapes suivantes) et la précision mais affecte la sensibilité.

L'objectif ici est de pouvoir identifier tous les k-mers communs entre la séquence en entrée et un ensemble de séquences en un temps indépendant de la taille de cet ensemble. Cela est rendu possible grâce à l'utilisation d'index (comme une table de hachage par exemple). A l'issue de cette étape, seules les séquences présentant un certain nombre de k-mers communs avec la séquence en entrée sont analysées par la suite.

(ii) Identification des régions de diagonales

On définit une *diagonale* comme l'ensemble des couples de positions (i, j) telles qu'il existe une constante $c \in \mathbb{Z}$, telle que $i - j = c$. Ainsi, deux hits (i, j) et (u, v) appartiennent à la même diagonale si et seulement si $i - j = u - v$.

Cette diagonale peut être visualisée dans une matrice « dot plot ». Pour deux séquences S_1 et S_2 de longueurs n_1 et n_2, on peut considérer une matrice de taille $n_1 \times n_2$ telle que toute coordonnée (i, j) de cette matrice prend la valeur 1 s'il existe un hit pour lequel la position i de S_1 est associée à la position j de S_2 et 0 sinon (voir Figure 2.31).

FastA procède à l'identification de régions de diagonales à forte similarité. Une région se caractérise par une triplet (i, j, l) correspondant à l'association de $S_1[i, i+l]$ et $S_2[j, j + l]$, elle débute et se termine nécessairement par un hit. Un score calculé comme la somme de scores position par position est associé à chaque région. Les 10 meilleurs régions sont conservées. On note que plusieurs régions peuvent appartenir à une même diagonale.

(iii) Affinage du score des régions

Chacune des dix régions sélectionnées au cours de l'étape précédente contient au moins un hit. Chacun de ces alignements contient à la fois des appariements (appartenant aux hits) et des mésappariements (appartenant aux fragments entre les hits) ; mais ils ne contiennent pas d'INDEL puisque les hits constitutifs de la région appartiennent à la même diagonale. Le score de chacune des 10 meilleures régions est calculé à nouveau mais à l'aide d'une matrice de scores appliquée aux hits de la diagonale. Pour chaque région, la sous-région de score maximal est conservée et notée `INIT1`.

(iv) Chaînage des régions initiales

Au cours de cette quatrième étape FastA tente de combiner les meilleures régions, soit celles calculées précédemment et telles que leur score soit supérieur à un seuil (qui peut être spécifié par l'utilisateur). Cette étape prend en compte les indels. Pour cela, FastA utilise un algorithme de chaînage de séquences tel que décrit à la

section 2.2.2, en intégrant un coût de gap proportionnel à l'espace séparant deux régions chaînées.

Par conséquent, deux régions initiales peuvent être chaînées si elles ne se chevauchent pas sur les séquences. Ces nouvelles associations intégrant les indel sont appelées `INITN`.

(v) Alignement avec Smith-Waterman

Pour terminer, en outre de `INITN`, FastA calcule un score d'alignement local alternatif. À partir de la région de meilleur score `INIT1`, FastA sélectionne une diagonale de largeur c centrée sur la diagonale comprenant cette région (par exemple, dans le cas des protéines avec une longueur de 2 pour les k-mers $c = 16$). FastA utilise alors une version de l'algorithme de Smith-Waterman restreint à la diagonale de largeur c. Le score d'alignement obtenu ici est noté `OPT`.

Même si l'algorithme FastA développé en 1988 a permis d'augmenter la rapidité des recherches de similarités dans les banques de données d'un facteur 10, cela est au détriment de la sensibilité des algorithmes de programmation dynamique capables d'identifier l'alignement local optimal entre deux séquences.

2.3.2 Algorithme BLAST

Développé au NCBI (National Center for Biotechnology Information) par Stephen Altschul en 1990, BLAST (Altschul and al, 1990), pour Basic Local Alignment Search Tool, est une méthode de recherche heuristique de régions similaires et d'alignement des régions homologues entre deux ou plusieurs séquences d'acides aminés ou de nucléotides. Cette méthode permet ainsi de retrouver rapidement dans une base de données les séquences similaires à une séquence requête, soit des séquences ayant des relations fonctionnelles ou évolutives et pouvant appartenir à une même famille de gènes.

Presque immédiatement après sa sortie, BLAST est devenu l'outil de recherche de séquences dans une base de données le plus utilisé. Cela s'explique en partie par sa rapidité et par le fait qu'il propose une liste ordonnée des résultats où chaque résultat est accompagné d'une estimation statistique de la significativité du résultat (soit la probabilité qu'un match de cette valeur ou de valeur supérieure soit retrouvée avec une séquence générée aléatoirement).

BLAST permet d'identifier les régions, sans gap, de plus haut score de similarité entre deux séquences, à l'aide d'une matrice de score sur l'alphabet considéré. Des alignements présentant quelques gaps peuvent être créés suite au chaînage des régions similaires identifiées ci-avant. Le filtre BLAST se base alors sur les définitions de *hit* (ou *segment pairs*), *hit local maximal* (ou LMSP, *Locally Maximal Segment Pairs*) et de *hit maximal* (ou MSP, *Maximal Segment Pairs*).

Définition 14 (hit local maximal & hit maximal)

Soient deux séquences S_1 et S_2. Un hit local maximal, est un hit dont le score d'alignement (sans gap) décroît si l'on étend ou si l'on réduit le hit d'un coté ou de l'autre. Un hit maximal est un hit de score maximal sur S_1 et S_2.

L'algorithme BLAST se déroule en trois étapes principales. Tout comme pour l'algorithme FastA les premières étapes ont pour but de localiser rapidement les régions de similarité sans gap entre la séquence en entrée et chacune des séquences de la base de données. Pour cela une analyse de similarité des résidus est réalisée en comparant tous les k-mers de chaque séquence. Les séquences présentant un hit maximal supérieur à une valeur seuil C (déterminée par un théorème [2]) sont conservées. Ainsi toutes les séquences de score supérieur à C sont considérées comme « significatives » et sont conservées. Blast présente également les séquences sans hits maximaux supérieur à C mais qui possèdent plusieurs hits maximaux dont la combinaison est statistiquement significative. De telles séquences sont identifiées par chaînage des hits maximaux.

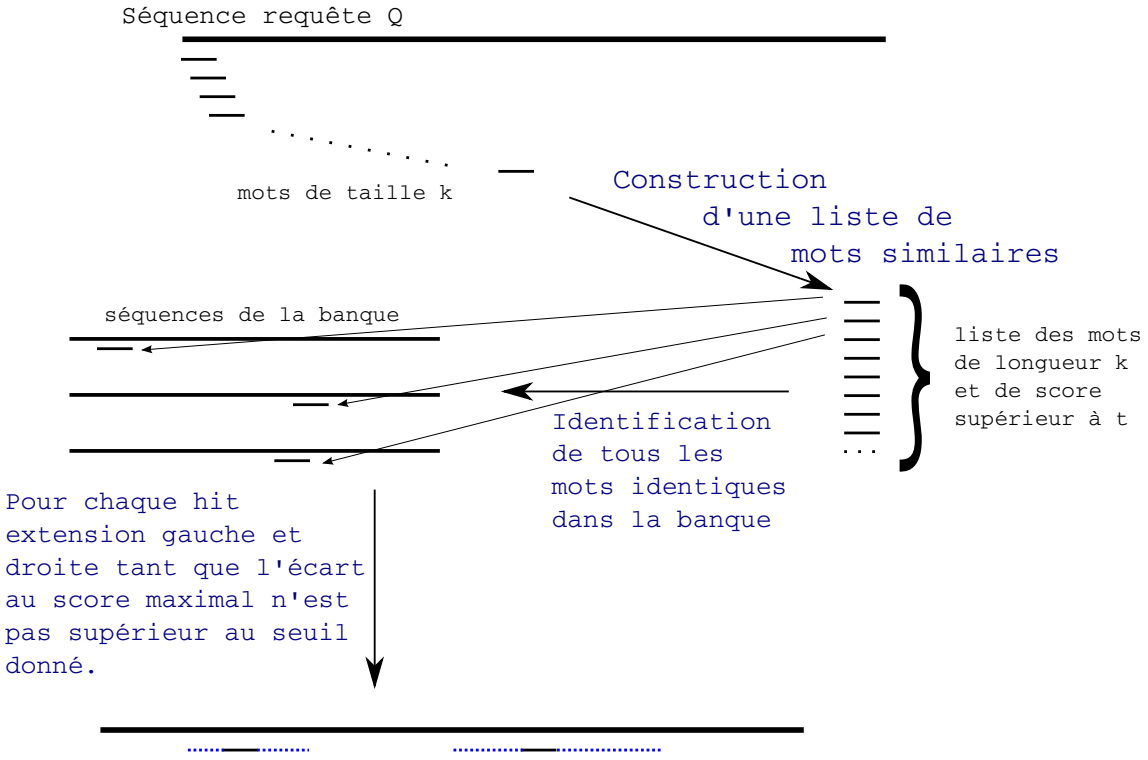


FIGURE 2.32 – *Principe de l'algorithme BLAST.*

2. Ce théorème identifie la plus petite valeur de C pour laquelle un MSP de score supérieur à C ne peut pas être obtenu de manière aléatoire dans aucune base de données.

(i) Prétraitement de la séquence requête-indexation

Dans un premier temps tous les k-mers de longueur k sur l'alphabet de la séquence sont générés (par exemple si k=2 sur l'alphabet des acides aminés il y a 441 mots et sur l'alphabet des acides nucléiques il y a 16 mots) et chaque k-mer de la séquence est comparé à l'ensemble exhaustif des k-mers générés. Chaque k-mer à chaque position de la séquence requête est associée la liste des k-mers ayant un score de similarité supérieur à une valeur seuil fixée, on parle de « mots voisins ».

(ii) Génération des *hits*

Suite à la première étape, chaque position de la séquence requête Q est représentée par la liste de mots voisins qui lui est associée. Comparer Q avec une séquence T de la banque de données consiste à comparer chaque k-mer de T avec chaque liste de toutes les positions de Q. Dès qu'un k-mer de T est similaire à l'un des k-mers d'une liste de Q les positions sont enregistrées comme un *hit*.

Tous les hits entre la séquence requête et chacune des séquences de la banque de données sont alors calculés.

(iii) Extension des *hits*

Afin de déterminer si chaque hit n'est pas inclu dans un hit ayant un score de similarité plus élevé, une extension sans gap de chacun de ces hits est réalisée par BLAST afin d'obtenir des hits maximaux locaux. Pour cela chaque hit est étendu de part et d'autre du k-mer. L'extension est interrompue dès que le score diminue d'une valeur w fixée par l'utilisateur par rapport au meilleur score d'alignement obtenu. Ainsi toute extension aboutissant à un segment de similarité de score supérieur ou égal à un seuil l, également fixé au préalable par l'utilisateur, est conservé et noté *HSP* (*High scoring Segment Pair*). Parmi tous les *HSP* identifiés entre deux séquences celui présentant le plus haut score de similarité est identifié comme étant le *MSP* (*Maximal Segment Pair*) ou hit maximal.

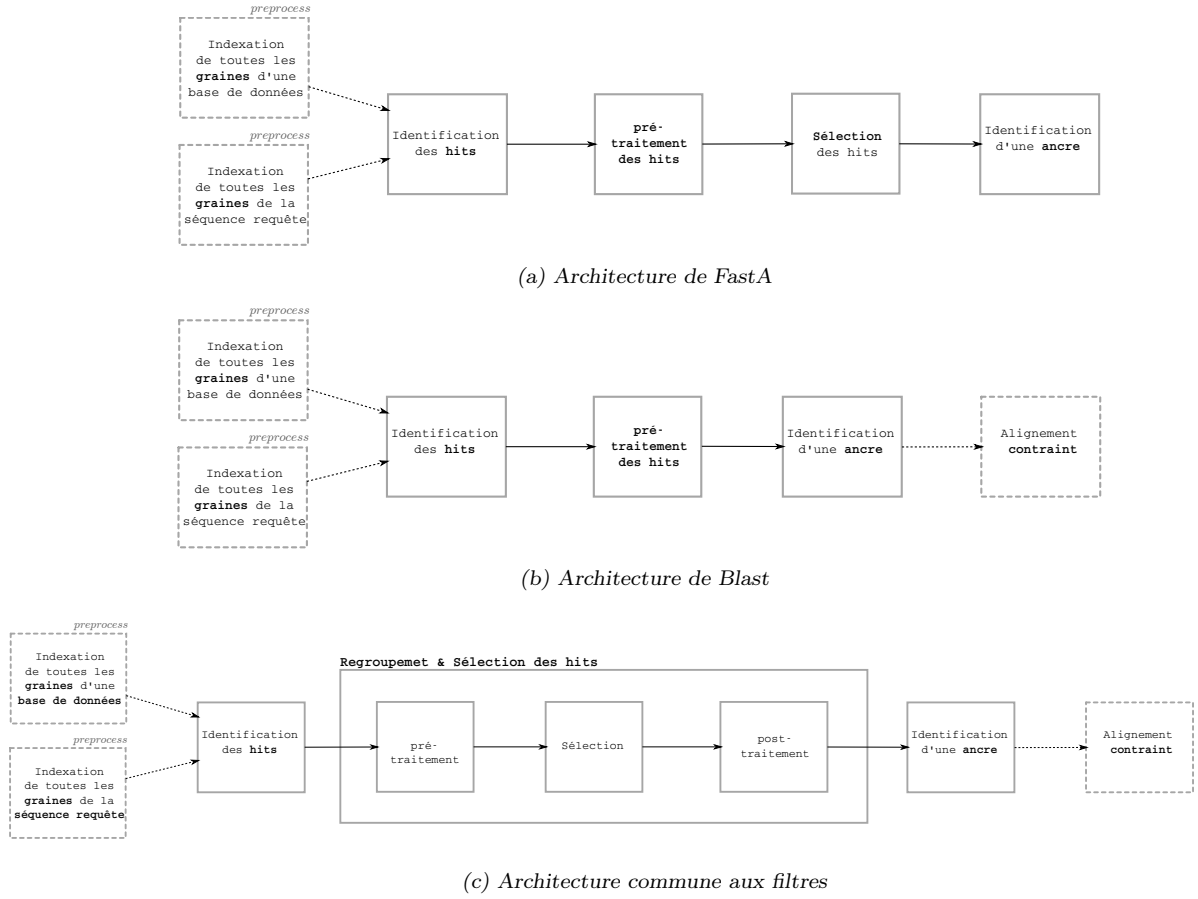
Ainsi l'heuristique BLAST recherche les segments les plus similaires entre une séquence requête et les séquences d'une base de données. Ne prenant en compte que les similarité sans gap, BLAST peut alors être moins sensible que FastA. Cependant une *p-value* peut être associée au score d'un hit maximal. Elle représente la probabilité qu'il existe au moins un hit maximal obtenu lors de la comparaison de deux séquences aléatoires de même longueur et même composition dont le score est supérieur ou égale à celui du hit maximal obtenu lors de la comparaison des vraies séquences.

(iv) Arrangement compatible des HSP

Dans le cas où deux séquences présentent plusieurs *HSP*, celles-ci sont ordonnées sur les séquences et seules les *HSP* compatibles sont alors sélectionnées pour calculer

la nouvelle valeur de score. Soit un *HSP* i de milieu (x_i, y_i) et un *HSP* j de (x_j, y_j), i et j sont convenablement ordonnés si $x_i < x_j$ et $y_i < y_j$ (Karlin and Altschul, 1993).

Conclusion



(a) Architecture de FastA

(b) Architecture de Blast

(c) Architecture commune aux filtres

FIGURE *2.33 – Architecture des deux filtres analysés et architecture commune extraite de leur analyse et servant de support au filtre décrit dans le Chapitre 3.*

Les deux filtres présentés, à savoir FastA et BLAST, présentent une architecture similaire (voir Figure 2.33). Leur approche consiste à identifier les motifs, ou graines, constitutifs de la séquence requête et de les comparer à ceux identifiés, possiblement au cours d'une étape préliminaire, au sein des séquences d'une base de données. Les séquences présentant des graines communes avec la séquence requête sont sélectionnées, on parle alors de hits. Ces hits peuvent alors être chainés ou combinés afin d'en extraire une ancre composée de hits compatibles. Ce type d'approche est celui suivi par le filtre présenté dans le Chapitre 3 qui suit.

Bibliographie

Altschul, S. and al (1990). Basic local alignment search tool. *Journal of molecular biology*, 215 :403–410.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., et al. (2003). The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1) :365–370.

Dayhoff, M. and Schwartz, R. (1978). A model of evolutionary change in proteins. In *In Atlas of protein sequence and structure*. Citeseer.

Fitch, W. (1966). An improved method of testing for evolutionary homology. *Journal of molecular biology*, 16(1) :9–16.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences : computer science and computational biology*. Cambridge University Press.

Henikoff, S. and Henikoff, J. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22) :10915–10919.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6) :341–343.

Höhl, M., Kurtz, S., and Ohlebusch, E. (2002). Efficient multiple genome alignment. *Bioinformatics*, 18(suppl 1) :S312–S320.

Jones, D., Taylor, W., and Thornton, J. (1992). The rapid generation of mutation data matrices from protein sequences. *Computer applications in the biosciences : CABIOS*, 8(3) :275–282.

Karlin, S. and Altschul, S. (1993). Applications and statistics for multiple high-scoring segments in molecular sequences. *Proceedings of the National Academy of Sciences*, 90(12) :5873–5877.

Levin, J., Robson, B., and Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS letters*, 205(2) :303–308.

Levitt, M. (1976). A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of molecular biology*, 104(1) :59–107.

Lipman, D. and Pearson, W. (1985). Rapid and sensitive protein similarity searches. *Science*, 227(4693) :1435–1441.

Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3) :443–453.

Pearson, W. and Lipman, D. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8) :2444–2448.

Risler, J., Delorme, M., Delacroix, H., and Henaut, A. (1988). Amino acid substitutions in structurally related proteins a pattern recognition approach : Determination of a new and efficient scoring matrix. *Journal of molecular biology*, 204(4) :1019–1029.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147 :195–197.

Chapitre 3

Filtrage de structures arborescentes et indexation rapide pour la recherche dans les bases de données

Actuellement la taille des bases de données croit rapidement, comme le montre entre autre l'évolution de la taille de la Rfam (Figure 1.24). Il est donc nécessaire de développer des algorithmes de filtrage des bases de données et de chaînage efficaces pour les ARN.

Dans un premier temps nous présenterons les différentes modélisations possibles de structures secondaires d'ARN. Puis, nous exposerons le problème d'édition et d'alignement d'arborescences permettant la comparaison de deux structures d'ARN. Avant de présenter notre filtre, nous décrirons la problématique de chainage dans les arborescences. Enfin, nous conclurons sur les performances de notre filtre.

3.1 Repliement, Modélisation et Comparaison de Structures Secondaires d'ARN

3.1.1 Modélisation, Notations et Définitions

Il existe différents niveaux d'analyse des séquences ARN. En effet, l'information contenue dans la structure primaire n'est pas toujours suffisante puisque deux ARN ayant des structures secondaires et tertiaires proches peuvent présenter des structures primaires différentes. Cependant, l'analyse de la structure tertiaire est généralement trop complexe, l'étude de la structure secondaire (se référer à la section 1.3.1.(iii)) apparaît alors comme la bonne alternative.

Repliement des ARN & Calcul de la Structure Secondaire

Trois principales méthodes permettent d'obtenir une structure secondaire d'un ARN. Cependant il faut garder en mémoire qu'un ARN possède souvent plus d'une

structure secondaire selon son environnement ou encore la fonction qu'il porte. Par exemple l'article (Nagel et al., 1999) présente des réarrangements conformationnels subtils se déroulant à dessein lors de la traduction d'un ARNm de *E. Coli*.

Approche expérimentale La première est une technique expérimentale qui utilise une méthode de cristallographie par diffraction aux rayons X ou de résonance magnétique nucléaire (Peattie and Gilbert, 1980). Même si elles sont plutôt précises ces méthodes sont longues et coûteuses et ne permettent d'obtenir la structure que de l'ARN d'intérêt dans les conditions étudiées puisque la conformation spatiale de l'ARN dépend également de son environnement. Ces méthodes expérimentales étant encore trop coûteuses en temps et en argent, des méthodes bioinformatiques de prédiction de structures sont développées. Cependant la prédiction de la structure tertiaire est un problème complexe bien que quelques travaux s'y essayent (Parisien and Major, 2008).

La seconde approche repose ainsi sur des algorithmes de prédiction dont les calculs s'effectuent sur une analyse de la structure primaire. Le problème consiste alors à identifier les bases qui s'apparient. Plusieurs algorithmes peuvent être dénombrés [1].

Approche *in silico* Un premier type permet de déterminer thermodynamiquement selon un modèle théorique de calcul d'énergie quelle est la structure secondaire la plus stable dans ce modèle, soit celle qui possède l'énergie libre minimum. Bien que cette approche permette d'obtenir de bons résultats pour les petits ARN, la structure prédite n'est souvent pas la structure adoptée pour les ARN de taille plus conséquente. En effet, la conformation spatiale adoptée par les ARN est celle dont l'énergie est proche de la conformation la plus stable et non celle dont l'énergie est minimale. C'est pour pallier ce problème que la plupart des programmes basés sur cette approche proposent un certain nombre de structures possibles dont l'énergie est proche de l'énergie minimale. Il faut alors choisir parmi toutes ces structures celle qui est la plus proche de la structure réelle. C'est à partir de cette étape qu'intervient la propriété biologique selon laquelle deux ARN ayant une fonctionnalité proche auront des structures similaires. Ainsi lorsque l'on recherche la structure secondaire d'un ARN, si l'on dispose de celle d'un second ARN ayant une fonction identique, il est alors possible de comparer chacune des structures proposées par le programme avec celle de l'ARN connu afin d'en extraire la structure secondaire la plus proche. Cependant le nombre de structures secondaires possibles augmente avec la taille de la séquence à replier.

Approche comparative Un autre type d'algorithme repose sur une analyse comparative d'un ensemble de séquences d'ARN dont on suppose qu'ils ont une même fonction (Perriquet et al., 2003; Touzet and Perriquet, 2004). On distingue alors deux

1. On note cependant que l'inclusion des pseudonoeuds dans les solutions rend le problème NP-complet (Lyngsø and Pedersen, 2000).

types d'approches comparatives selon que les séquences en entrée aient été alignées au préalable ou pas. Lorsque un alignement préalable des séquences en entrée est nécessaire, comme par exemple avec *RNA-alifold* (Lorenz et al., 2011) ou *RNAz* (Hofacker and Stadler, 2010), la recherche de la structure secondaire commune se fait à partir de la détection des covariations de l'alignement fourni.

Lorsque aucun alignement n'est disponible, deux approches peuvent être employées : soit une approche dérivée de l'algorithme de Sankoff, comme par exemple *Foldalign* (Havgaard et al., 2007), soit une approche heuristique, comme par exemple *CARNAC* (Touzet and Perriquet, 2004), au cours de laquelle on calcule, dans chaque séquence, l'ensemble des tiges-boucles candidates puis on recherche pour chaque paire de séquences les régions conservées et on sélectionne les tiges compatibles pour terminer par un alignement simultané des tiges alignables.

Afin de comparer des structures secondaires il est nécessaire de trouver un formalisme qui permette de rendre compte des différents motifs qui composent l'ARN mais aussi de comparer rapidement deux structures entre elles.

(i) Représentation des Structures Secondaires

De nombreux modèles ont été définis afin de pouvoir représenter et comparer des structures secondaires d'ARN. Bien que par la suite seules les représentations arc-annotée et arborescentes seront utilisées, nous présentons brièvement l'ensemble des modélisations usuelles utilisées dans la littérature.

Projection planaire La structure secondaire d'ARN a la particularité d'être une conformation dans le plan. Elle est donc facilement représentable dans le plan d'une feuille (voir Figure 3.36 (a)). La projection planaire de la structure secondaire d'un ARN correspond ainsi à la projection en deux dimensions de cette structure. Cette représentation permet d'identifier aisément les différents composants de cette structure.

Séquence arc-annotée L'utilisation de la structure de *séquence annotée par des arcs* ou *séquence arc-annotée* pour modéliser la structure secondaire d'ARN a été introduite par Evans (Evans, 1999). Dans ce modèle la structure primaire est représentée sur une ligne par une séquence et est augmentée par des arcs représentés du même coté de la ligne et qui relient deux symboles entre eux (voir Figure 3.34). Ces arcs modélisent les liaisons entre les deux nucléotides appariés [2]. Cette représentation fait suite à la « Mountain representation » (Hogeweg and Hesper, 1984) dans laquelle les nucléotides appariés sont reliés par des lignes de niveau (les deux nucléotides d'une paire sont alors à chaque extrémité de cette ligne de niveau) et les nucléotides non appariés sont représentés sur le même niveau que le nucléotide

2. Dans cette représentation les pseudo-noeuds sont facilement identifiables par des arcs qui se croisent.

précédant. La superposition de ces lignes de niveau définit alors une montagne ou une chaine de montagne représentatant la structure secondaire de l'ARN.

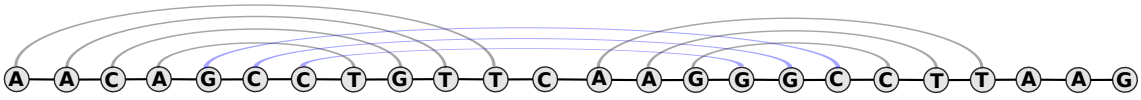


FIGURE 3.34 – *Représentation arc-annotée d'une séquence ARN. Les arcs symbolisent les liaisons entre les nucléotides. Les arcs bleus représentent des pseudo-noeuds.*

Graphe de corde Le graphe de corde est une représentation circulaire de la séquence arc-annoté. Pour cela la première et la dernière base sont connectées, la structure primaire constitue alors la périphérie du cercle et la structure secondaire les cordes de ce cercle [3] (voir Figure 3.35).

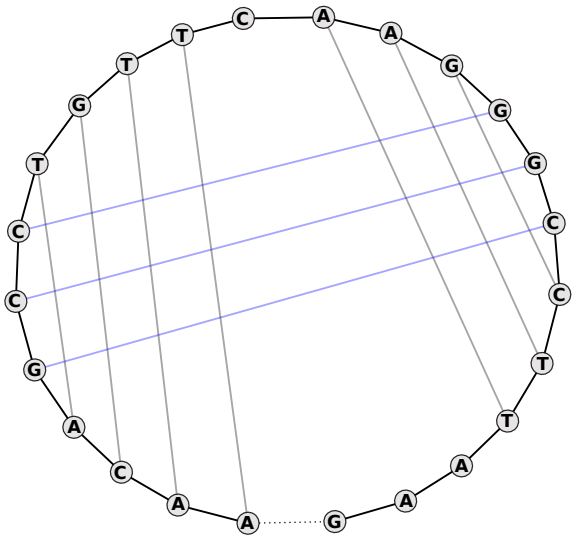


FIGURE 3.35 – *Représentation en graphe de corde d'une séquence ARN. Les cordes symbolisent les liaisons entre les nucléotides. Les cordes bleus représentent des pseudo-noeuds.*

3. Les pseudo-noeuds apparaissent ici sous la forme de cordes qui se croisent.

Mot de Motzkin La structure secondaire d'ARN dépourvue de pseudo-noeuds peut être codée par des mots de Motzkin. Pour cela toute base i appariée à une base j telle que $i < j$ est associée au symbole « (» et la base j au symbole «) » et toute base i non appariée est associée au symbole « . ». Le format de prise en charge des structures secondaires du package Vienna (Lorenz et al., 2011) utilise ce format de structure parenthésée auquel on donne alors le nom de format Vienna ou format dot-bracket [4](qui est une classe particulière de séquences arc-annotées imbriquées) :

```
>NomARN
aacccaacccaacccaagggacccaagggaagggaagggaa
..(((..(((.......))).(((.......)))..)))..
```

Arborescences Ordonnées et Graphes Parmi les représentations les plus courantes on remarque principalement celle introduite par Zuker et Sankoff (Zuker and Sankoff, 1984) : l'arborescence ordonnée étiquetée. Cette structure secondaire sous forme d'arbre est obtenue en associant chaque base non appariée à une feuille et chaque paire de bases à un noeud interne (voir Figure 3.36). Si l'on introduit les pseudonoeuds, les bases impliquées sont alors reliées entre elles faisant passer cette représentation d'un arbre à un graphe. Dans cette représentation si les premières bases ne sont pas appariées il est possible de rajouter un premier noeud afin qu'il serve de racine pour enraciner l'arbre. Cependant ce n'est pas obligatoire et si l'arbre n'est pas enraciné le graphe de la structure secondaire d'ARN obtenu est une forêt.

D'autres représentations arborescentes ont également été développées. Dans (Shapiro and Zhang, 1990) les auteurs proposent une représentation dans laquelle les noeuds symbolisent les éléments de structures secondaires et les arcs qui les relient symbolisent les hélices ou les tiges (voir Figure 3.36 (c)). Les noeuds sont alors étiquetés selon la structure qu'il représentent : R pour la racine, M pour une multiboucle, B pour un renflement, I pour une boucle interne et H pour une boucle terminale. Une autre représentation est celle de RNAforester (Hochsmann et al., 2003; Schirmer and Giegerich, 2013) qui est une modélisation étendue de celle de Zuker et Sankoff (Zuker and Sankoff, 1984) dans laquelle les paires de bases sont représentées par un noeud interne étiqueté P auquel sont connectés deux noeuds fils, un à droite et un à gauche pour chaque base appariée (voir Figure 3.36).

Pour nos travaux deux représentations ont été retenues et utilisées alternativement : la séquence arc-annotée et le format d'arbre ordonné. Ainsi pour toute structure secondaire d'ARN sans pseudonoeud, il est possible de définir une séquence arc-annotée et un arbre enraciné et ordonné modélisant cette structure.

4. Certains pseudonoeuds peuvent être introduit en encodant les bases connectant des boucles par les symboles « [» et «] ».

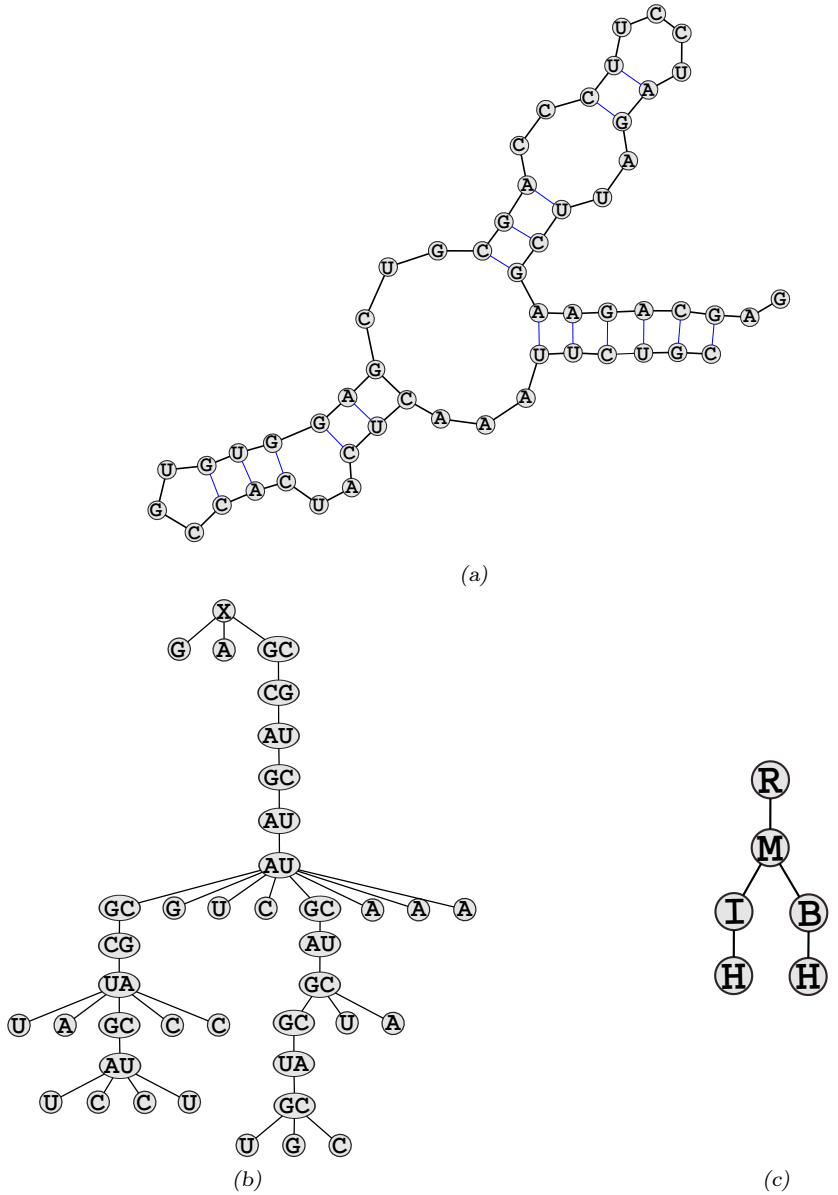


FIGURE 3.36 – *Diverses représentations en arborescence d'une séquence ARN.(a) Représentation planaire. (b) Représentation de Zuker (Zuker and Sankoff, 1984). (c) Représentation de Shapiro (Shapiro and Zhang, 1990)*

Définition 15 (Séquence Arc-annotée & Arbre Enraciné Ordonné)

Soit une structure secondaire d'ARN. Dans le cas où le premier et le dernier nucléotides de l'ARN ne forment pas une liaison, on ajoute deux nucléotides fictifs formant une liaison au début et à la fin de cet ARN.

On définit la séquence arc-annotée $A = (S, P)$ associée à cette structure par une séquence S de longueur n modélisant la structure primaire de l'ARN et un ensemble

P de couples de positions sur S tels que $(i, j) \in P$ s'il existe une liaison nucléotidique entre $S[i]$ et $S[j]$.

On définit également un arbre enraciné T tel que chaque paire de nucléotides de la structure secondaire d'ARN est associée à un noeud interne de l'arbre et chaque nucléotide non apparié est associé à une feuille de T. Si le noeud v est fils du noeud u, alors u encode pour une liaison d'une hélice x de la structure :

- *Soit u n'est pas la dernière liaison de x, alors v est un noeud interne et encode pour la liaison directement consécutive à u dans x.*
- *Soit u est la dernière liaison de x, alors :*
  - *x se termine par une boucle, v est une feuille et encode pour un des nucléotides de cette boucle,*
  - *x se termine sur une boucle interne ou un renflement, v est un noeud interne et encode pour la première liaison de l'hélice suivante ou v est une feuille et encode pour un des nucléotides de la boucle ou du renflement,*
  - *x se termine par une boucle multiple, v est un noeud interne et encode pour la première liaison d'une des hélices de la boucle ou v est une feuille et encode pour un des nucléotides de la boucle.*

On définit le rang d'un sommet de T comme : l'indice du nucléotide dans le cas d'une feuille et dans le cas d'un noeud interne, le plus petit indice des nucléotides composant la liaison. Soit v et w deux fils de u, alors v est à gauche de w si le rang de v est inférieur au rang de w.

Finalement, on constate que A et T sont des structures duales et qu'il existe par conséquent une bijection entre les noeuds internes de T et les couples de P. Il existe une injection entre les feuilles de T et les positions sur S. Enfin, il existe une surjection associant toute position de S à un sommet de T.

Ces deux représentations facilement interchangeables présentent toutes deux un intérêt pour la comparaison des structures secondaires. En effet, nous allons voir que la notion de graines se définit plus aisément sur une séquence arc-annotée tandis que le chaînage des hits s'exprime plus simplement sur les arbres.

3.1.2 Problème de Comparaison d'ARN Deux à Deux

Plusieurs méthodes de comparaisons d'arborescences ou de séquences arc-annotées deux à deux ont été définies (Blin and Touzet, 2006; Bille, 2005; Demaine et al., 2009). Ici nous focaliserons cette étude sur deux de ces méthodes : l'édition et l'alignement de deux arborescences.

(i) Édition d'Arborescences

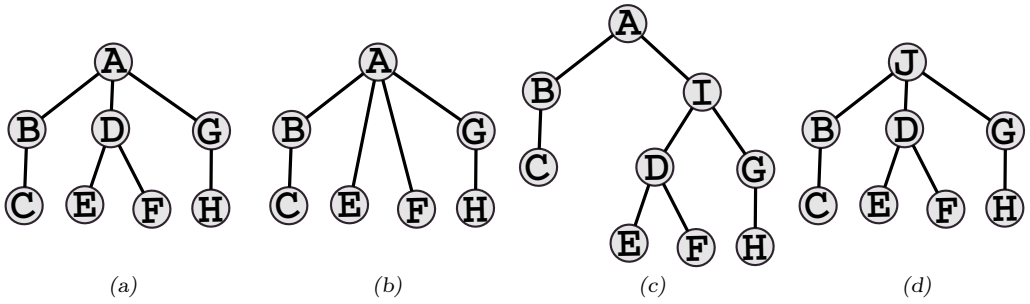


FIGURE 3.37 – *Différentes opérations d'édition appliquées aux arborescences. (a) arborescence originale. (b) délétion du noeud étiqueté D. (c) insertion du noeud étiqueté I. (d) substitution du noeud étiqueté A par le noeud étiqueté J.*

L'édition dans les arborescences repose sur trois opérations d'édition, appliquées aux noeuds de cette arborescence :

- la *délétion*
- l'*insertion*
- la *substitution*

Lors de la *délétion* du noeud u, les fils de u conservent leur ordre et deviennent les fils du père de u, le fils le plus à gauche de u va à droite du frère gauche de u. Le fils le plus à droite de u va à gauche du frère droit de u. L'*insertion* d'un noeud est alors l'opération symétrique à la *délétion*. Lors de l'*insertion* d'un noeud u comme fils de v, celui-ci devient le père d'un certain nombre de fils consécutifs de v. Enfin la *substitution* d'un noeud consiste à modifier l'étiquette de ce noeud.

On définit un chemin d'édition valide entre deux arbres T_1 en T_2 comme suit :

Définition 16 (Chemin d'Édition Valide)

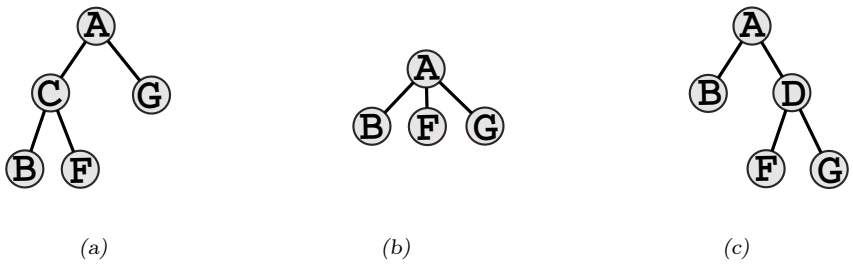
Soient T_1 et T_2 deux arborescences enracinées et ordonnées. Soit E une suite d'opérations d'édition. E est un chemin d'édition valide de T_1 en T_2, si pour tout sommet de T_1, il existe une unique opération de E appliquée à ce sommet et, en appliquant les opérations de E à T_1, on obtient l'arbre T_2.

Si l'on pondère chaque opération d'édition, la distance d'édition se définit comme :

Définition 17 (Distance d'Édition)

Soient T_1 et T_2 deux arborescences enracinées et ordonnées. Soit une fonction de coût c qui associe un coût à chaque opération d'édition. Ce coût est strictement positif lors de la substitution de sommets différents et nul pour la substitution de sommets identiques. Ce coût est strictement positif et de même valeur pour l'insertion et la délétion.

Pour tout chemin d'édition E, on associe un coût $c(E) = \sum_{e \in E} c(e)$. Soit l'ensemble des chemins d'édition valides de T_1 à T_2, la distance d'édition entre T_1 et T_2 est définie comme le plus petit coût parmi les coûts des chemins valides.



(a) (b) (c)

Figure 3.38 – *Différentes opérations d'édition appliquées à deux arborescences. (a) première arborescence. (b) Délétion du noeud C. (c) Insertion du noeud D-deuxième arborescence.*

À partir d'un chemin d'édition valide entre deux arbres, il est possible de construire une association entre leurs noeuds.

Définition 18 (Mise en correspondance)

Soient T_1 et T_2 deux arborescences et soit E un chemin d'édition valide. On définit la mise en correspondance $\mathcal{M}$ associée à E comme un ensemble des couples de sommets sur $T_1 \times T_2$ tel que $(t_1, t_2) \in \mathcal{M}$ s'il existe une opération de substitution dans E qui associe t_1 à t_2. Par construction, $\forall (v_i, v_j) \in \mathcal{M}$ et $(v'_i, v'_j) \in \mathcal{M}$:

-unicité- $v_i = v'_i$ si et seulement si $v_j = v'_j$.

-ancestralité- v_i est un ancêtre de v'_i si et seulement si v_j est un ancêtre de v'_j.

-ordre- v'_i est à droite de v_i, si et seulement si v'_j est à droite de v_j.

Le problème du calcul de la distance d'édition entre deux arbres enracinés ordonnés se résoud par un algorithme de programmation dynamique. La meilleure complexité obtenue à ce jour est de $O(n^3)$, pour deux arbres de taille n (Demaine et al., 2009).

(ii) Alignement d'Arborescences

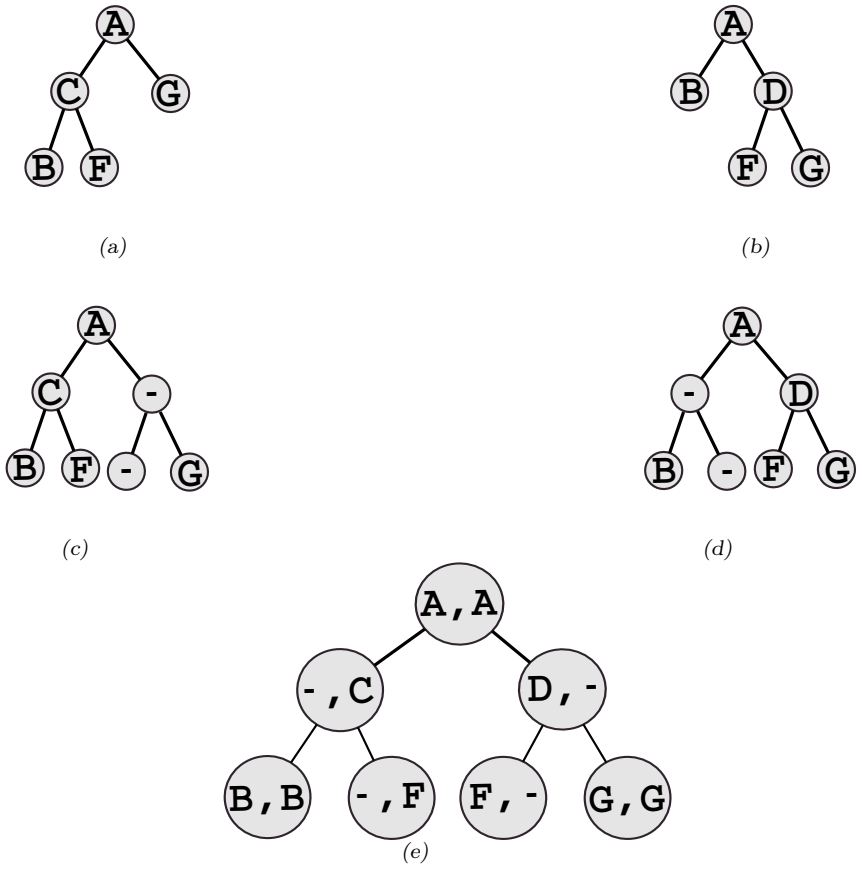


FIGURE *3.39 – Différentes opérations d'alignement appliquées à deux arborescences. (a) et (b) arborescences originales. (c) et (d) Insertion des noeuds. (e) Super arborescence commune.*

L'alignement de deux arbres enracinés ordonnés se définit de la même façon que pour les séquences. À partir de deux arbres T_1 et T_2, on insère des noeuds étiquetés '-' jusqu'à l'obtention de deux arbres T'_1 et T'_2 isomorphes [5]. L'alignement correspond à l'arbre $T'_{1,2}$ dont les étiquettes des noeuds sont des couples correspondant aux étiquettes respectives de T'_1 et T'_2 :

5. De même topologie mais dont les étiquettes sont possiblement différentes.

Définition 19 (Alignement de deux Arbres Enracinés Ordonnés)

Soient deux arborescences enracinées ordonnées T_1 et T_2 dont les noeuds sont étiquetés sur un alphabet Σ, un arbre enraciné ordonné $T'_{1,2}$ dont les noeuds sont étiquetés par des couples de symboles sur l'alphabet $\{\Sigma \cup' -'\}$ est un alignement valide de T_1, T_2 si :

- *il n'existe pas de noeud de $T'_{1,2}$ étiqueté $('-','-')$*
- *soit T'_1 l'arbre isomorphe à $T'_{1,2}$ dont les noeuds sont étiquetés par la première composante des couples des étiquettes des noeuds de $T'_{1,2}$. Si l'on supprime[6] l'ensemble des noeuds de T'_1 étiquetés '-', on obtient T_1.*
- *soit T'_2 l'arbre isomorphe à $T'_{1,2}$ dont les noeuds sont étiquetés par la deuxième composante des couples des étiquettes des noeuds de $T'_{1,2}$. Si l'on supprime[6] l'ensemble des noeuds de T'_2 étiquetés '-', on obtient T_2.*

Maintenant, si l'on considère l'ensemble possible des alignements entre deux arbres, on peut associer un score à chacun d'entre eux afin de définir un alignement optimal :

Définition 20 (Alignement Optimal de deux Arbres Enracinés Ordonnés)

Soient deux arborescences enracinées ordonnées T_1 et T_2 dont les noeuds sont étiquetés sur un alphabet Σ, soit l'ensemble $\mathcal{T}$ des alignements valides entre T_1 et T_2, soit enfin une fonction score : $\{\Sigma \cup' -'\}^2 \to \mathbb{R}$ qui pour chaque couple d'étiquettes associe une valeur. Pour chaque alignement T de $\mathcal{T}$, on définit son score comme :

$$Score(T) = \sum_{u \in T} score(label(u))$$

Un alignement $T \in \mathcal{T}$ est optimal s'il n'existe pas d'alignement $T' \in \mathcal{T}$ tel que $Score(T') > Score(T)$. Le score de cet alignement correspond au score d'alignement de T_1 et T_2.

On observe, comme l'illustrent les Figures 3.39 et 3.38, que la comparaison de deux arborescences par alignement ou par édition n'est pas équivalente. L'alignement peut être vu comme un cas particulier de l'édition dans lequel toutes les insertions précèdent les délétions.

Le meilleur algorithme connu à ce jour pour résoudre ce problème à une complexité de $O(n^3)$ où n est la taille des deux arbres alignés (Jiang et al., 1995; Blin and Touzet, 2006).

6. La suppression se fait suivant la même définition que l'opération de suppression de l'édition

(iii) Édition et Alignement de Séquences Arc-Annotées

En raison de la dualité entre les structures arborescentes et les structures arc-annotées, tout comme il est possible de comparer de deux manières différentes deux arbres enracinés ordonnés, il est possible de comparer de deux manières différentes des séquences arc-annotées. L'ensemble des opérations d'édition et d'alignement décrites sur les arborescences sont transposables aux séquences arc-annotées. Dans ce cas les séquences arc-annotées sont d'une classe particulière dite *imbriquée* (Evans, 1999). L'édition de deux séquences arcs-annotées de cette classe avec les trois opérations d'édition à savoir, la substitution d'arc (bases inférentes comprises) ou de base libre, la délétion d'arc (bases inférentes comprises) ou de base libre et l'insertion d'arc (bases inférentes comprises) ou de base libre, est strictement équivalent au cas des arbres et se fait en $O(n^3)$. Il en va de même pour l'alignement de séquences arc-annotées imbriquées et dont l'alignement est lui-même une séquence arc-annotée imbriquée qui peut être réalisé en $O(n^4)$.

Cependant plusieurs opérations supplémentaires peuvent s'y ajouter puisque, contrairement aux noeuds des arbres qui prennent simultanément en compte la structure (type du noeud) et la séquence (étiquette du noeud), les entités symbolisant la structure (arc) et la séquence sont distinctes. Ainsi, nous pouvons ajouter des opérations telles que la délétion d'arc (sans délétion des bases inférentes) ou l'altération d'arc. Selon l'ensemble d'opérations considérées et la classe des séquences arc-annotées en entrée, le problème peut devenir NP-complet (Blin and Touzet, 2006). Il en va de même pour l'alignement en fonction de la classe des séquences arc-annotées en entrée ainsi que de la classe de la séquence arc-annotée correspondante à l'alignement.

(iv) Comparaison de Deux ARN Sans Structure (LocARNA)

Dans le cas où on ne dispose pas des structures secondaires, il est possible d'aligner deux séquences ARN tout en calculant simultanément une structure secondaire compatible avec les deux séquences ARN.

LocARNA (Local alignment of RNA) (Heyne et al., 2009) est une méthode basée sur une variante de l'algorithme de Sankoff (1985) permettant d'aligner et de replier simultanément deux séquences ARN. *LocARNA* prend en entrée des séquences ARN non alignées et calcule, séparément pour chacune d'elles, les probabilités d'appariement des bases avec *RNAfold*.

Soient deux séquences S_1 et S_2, *LocARNA* commence par calculer, avec *RNAfold*, les matrices de probabilités d'appariement de bases, P^1 et P^2, de chaque séquence. Aligner S_1 et S_2 consiste à maximiser un score d'alignement à partir des score d'alignement des bases appariées et non appariées. La complexité en temps de cette approche est $O(n^4)$ et $O(n^2)$ en espace, où n est la taille des séquences en entrée. L'algorithme original de Sankoff (1985) présente une complexité en temps de $O(n^6)$.

3.2 Chainage dans les Arborescences

Nous venons de voir qu'il existe diverses méthodes de comparaison de deux ARN. Ces méthodes présentent des complexités (au moins cubique) ne permettant pas un passage à l'échelle. Cependant, pour chacune de ces méthodes il est possible d'injecter des contraintes, c'est-à-dire de forcer la mise en correspondance de certaines bases des ARN en entrée, et ainsi de réduire de façon significative le temps de calcul. Le problème consiste alors à établir ces contraintes. Pour cela, de même que pour les filtres sur les séquences, on va identifier de courts motifs communs entre les deux ARN. L'ensemble de ces motifs ne forme pas nécessairement un ensemble de contraintes valides. C'est pourquoi il est nécessaire d'extraire de cet ensemble de motifs un sous-ensemble cohérent. Ce qui constitue le coeur du problème de chainage que nous allons présenter.

3.2.1 Définitions, Préliminaires et Établissement du Problème

Avant d'aborder le problème du chainage entre deux arborescences, il est nécessaire d'introduire un certain nombre de notions afin de pouvoir établir le problème qui nous intéresse, à savoir le problème de chainage entre deux arborescences ordonnées enracinées.

(i) Définitions et Préliminaires

Soit T une arborescence enracinée ordonnée de taille n. On note $T[i]$, le noeud de T à l'indice i selon un parcours post-fixe. Ainsi $T[n-1]$ désigne la racine de T, $T[0]$ la feuille la plus à gauche de l'arbre. On note T_i le sous-arbre de T ayant pour racine $T[i]$.

Comme pour le chaînage en séquence, le chaînage dans les arborescences ordonnées enracinées repose sur l'identification de graines communes à deux arborescences ordonnées enracinées T_1 et T_2 à comparer. De telles graines définissent, comme dans les séquences, un *hit*. Ainsi une graine représente une sous partie d'un arbre, et un hit une mise en correspondance entre deux sous parties de deux arbres. Nous commençons par introduire un modèle de graine pour les arborescences.

Définition 21 (Graine dans une Arborescence)

Soit T une arborescence enracinée ordonnée. Une graine $G = (g_0, \ldots, g_k)$ est définie comme un ensemble de noeuds de T vérifiant certaines contraintes :

- *On suppose que $g_0 < g_1 < \cdots < g_k$*
- *Si les noeuds de G forment une sous-structure connectée de T, alors G est un arbre interne de T dont la racine est g_k.*
- *Si les noeuds de G ne forment pas une sous-structure connectée de T, alors il existe une partition de G en p arbres internes $G^1 \ldots G^p$ telle que pour $i \in [2, p]$ la racine de G^{i-1} est le frère gauche de la racine de G^i. Dans ce cas, on parlera de forêt interne compacte de T.*

On remarque que l'arbre interne est un cas particulier de la forêt interne où $p = 1$. On appellera racine de la graine G le noeud de celle-ci d'indice le plus élevé, c'est-à-dire la racine de l'arbre le plus à droite de la forêt interne. On appellera bords de la graine G l'ensemble $B(G) \subset G$ tel que pour tout noeud u de cet ensemble, soit u est une feuille dans G, soit il existe un fils de u dans T qui n'appartient pas à G.

Il nous est maintenant possible d'introduire la définition d'un hit à savoir la mise en correspondance de deux graines entre deux arbres.

Définition 22 (Hit entre deux Arborescences)

Soient T_1 et T_2 deux arborescences enracinées ordonnées, soient G_1 et G_2 deux graines de T_1 et T_2 respectivement. Le couple (G_1, G_2) forme un hit si et seulement si :

- *le nombre de sous-arbres internes qui composent G_1 est identique à celui de G_2.*
- *le nombre de noeuds qui composent les bords de G_1 et G_2 est identique.*
- *soient $B_1(G_1) = (b_1^1, b_1^2, \ldots, b_1^\ell)$ et $B_2(G_2) = (b_2^1, b_2^2, \ldots, b_2^\ell)$, alors l'ensemble $E = \{(b_1^1, b_2^1), \ldots, (b_1^\ell, b_2^\ell)\}$ forme une mise en correspondance valide entre T_1 et T_2 (voir Définition (i)).*

On note $\mathcal{E}_{G_1,G_2}$ la mise en correspondance composée de l'ensemble E ci-dessus ainsi que de l'ensemble des mises en correspondances entre les racines des sous-arbres internes qui composent G_1 et G_2. $\mathcal{E}_{G_1,G_2}$ définit une mise en correspondance valide entre T_1 et T_2

On peut observer que, de même que pour les séquences, deux graines n'ont pas a être de même taille pour former un hit. Étant donné maintenant un ensemble de hits, nous pouvons définir une chaine de hits valide comme suit :

Définition 23 (Chaine de Hits Valide)

Soient T_1 et T_2 deux arborescences enracinées ordonnées et $h = (G_1, G_2)$ et $h' = (G'_1, G'_2)$ deux hits sur T_1 et T_2. Deux hits h et h' sont chainables si et seulement si :
- *$G_1 \cap G'_1 = \emptyset$ et $G_2 \cap G'_2 = \emptyset$.*
- *$\mathcal{E}_h \cup \mathcal{E}_{h'}$ est une mise en correspondance valide entre T_1 et T_2.*

Soit maintenant $\mathcal{H}$ un ensemble de hits sur T_1 et T_2, $\mathcal{H}$ est une chaine valide si tous les hits qui la composent sont chainables deux à deux.

On peut maintenant définir le problème de chaînage dans les arborescences comme suit.

(ii) Problème de Chaînage Maximum dans les Arborescences

Le problème de chaînage ainsi étudié consiste à calculer la chaine de hits non chevauchants et de score maximal entre deux structures secondaires arborescentes données (Allali et al., 2012) ou « Maximal Chaining Problem ». Ce problème est illustré par la Figure 3.40.

Définition 24 (Problème de Chaînage Maximal dans une Arborescence)

Soient T_1 et T_2 deux arborescences enracinées ordonnées, $\mathcal{H}$ l'ensemble des hits entre T_1 et T_2 et une fonction score : $\mathcal{H} \to \mathbb{R}$ qui associe un score à chaque hit. Le problème de chaînage maximal consiste à établir la chaine $\mathcal{A} \subset \mathcal{H}$ de score maximal telle que :
- *le score de $\mathcal{A}$ est donné par la somme $\sum_{h \in \mathcal{A}} score(h)$.*
- *$\mathcal{A}$ est une chaîne valide sur T_1 et T_2.*
- *il n'existe pas de chaîne $\mathcal{A}'$ valide sur T_1 et T_2 dont le score est supérieur au score de $\mathcal{A}$.*

Plus généralement, on définit le problème de chaînage maximal (MCP) par :

$$MCP(T_1, T_2, \mathcal{H}) = max\{\sum_{h \in \mathcal{A}} score(h); \mathcal{A} \subset \mathcal{H}\}$$

3.2.2 Deux Algorithmes de Chainage 2D dans les Arborescences

Tout comme pour le chainage en séquence discuté à la section 2.2.2, deux principaux types d'algorithmes de chainage sur les arborescences ont été proposés : un algorithme de chainage par balayage dans les arborescences (Allali et al., 2012) et un algorithme de chainage par programmation dynamique dans les arborescences (Heyne

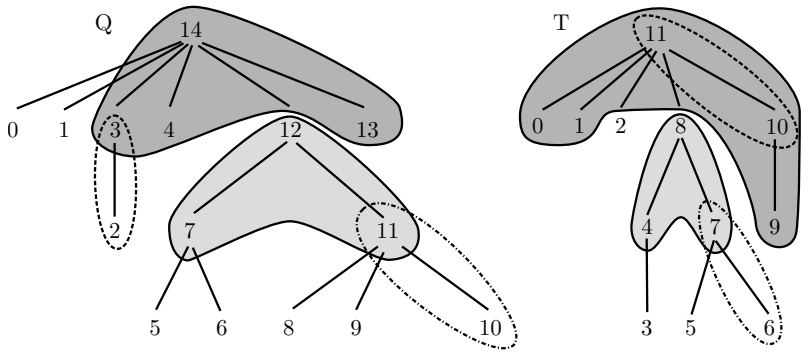


FIGURE 3.40 – *Exemple, extrait de l'article d'Allali et al. (2012), du problème de chaînage maximal dans deux arborescences présentant 6 hits :* $P_0 = \{(2,10),(3,11)\}$, $P_1 = \{(6,3)\}$, $P_2 = \{(9,5)\}$, $P_3 = \{(10,6),(11,7)\}$, $P_4 = \{(7,4),(11,7),(12,8)\}$, $P_5 = \{(3,1),(13,9),(14,11)\}$. *Si le score d'un hit est donné par le nombre de noeuds composant le hit, une chaîne optimale est ici composée de* $\{P1, P2, P4, P5\}$ *avec pour score* 8.

et al., 2009).

(i) Algorithme de Programmation Dynamique de Chainage 2D dans les Arborescences

Récemment, Heyne et al. (2009) ont introduit le problème de chaînage sur les structures en séquences arc-annotées qu'ils résolvent, sous certaines restrictions, avec un algorithme de programmation dynamique. Pour cela ils considèrent des graines qui sont définies comme des régions maximales de similarité exacte, Exact Pattern Matching (EPM), en séquence et en structure. À partir de l'identification de ces EPM, ils appliquent leur algorithme de programmation dynamique comme suit :

- Détection d'une chaîne maximale d'EPM entre deux structures secondaires d'ARN.
- Analyse de chaque région entre deux EPM successives à l'aide d'un algorithme de calcul de distance d'édition, afin d'accélérer le processus de comparaison.

Heyne et al. (2009) est le premier article abordant la question du chaînage dans des arbres. La méthode de programmation dynamique qu'ils emploient est à la fois différente et plus simple que les approches habituellement employées pour ce genre de problème dans les séquences. Cette méthode utilise le même principe que celui détaillé dans la section 2.2.2 du Chapitre 2 concernant le chaînage dans les séquences. On remarque que lorsque cet algorithme est employé sur des séquences arc-annotées ne présentant aucun arc, c'est-à-dire sur des séquences, on peut démontrer que cet algorithme présente une complexité en $O(n^2 + k)$, où k est le nombre d'EPM et n la taille des séquences, plus élevée que celle du meilleur algorithme de chaînage en séquences connu (Heyne et al., 2009; Allali et al., 2012).

Cela soulève alors la question de l'implémentation d'un algorithme plus efficient aussi bien d'un point de vue théorique que pratique. C'est dans cette optique qu'a été élaboré le second algorithme de chainage dans les séquences se basant, comme l'algorithme en séquence, sur le balayage des arborescences.

(ii) Algorithme par Balayage de Chainage 2D dans les Arborescences

Cet algorithme présente une solution par balayage des arborescences (Allali et al., 2012) qui résout le problème avec une complexité inférieure à l'algorithme de programmation dynamique (Heyne et al., 2009) tant que le nombre de hits à analyser est « raisonnable », c'est-à-dire inférieur à $n_1 \times n_2$, où n_1 et n_2 sont la taille des arbres à comparer.

Le problème de chaînage maximal présente alors une complexité en $O(|\mathcal{H}| \log(|\mathcal{H}|) + k|\mathcal{H}| \log(k))$ en temps, où $|\mathcal{H}|$ est le nombre total de bords des hits de $\mathcal{H}$, et $O(k \times |\mathcal{H}|)$ en espace. En particulier, on observe que le temps de calcul est indépendant de la taille des arborescences support et qu'avec cet algorithme le problème de chaînage en séquence présente la même complexité que le meilleur algorithme connu de chaînage en séquences (Ohlebusch and Abouelhoda, 2006), à savoir $O(k \log(k))$ en temps et $O(k)$ en espace.

3.3 *RNA-unchained* : Un Filtre pour la Comparaison d'Arborescences

L'algorithme de chainage établi dans (Allali et al., 2012) est basé sur les structures arborescentes mais comme explicité dans la section 3.1.1(i) il est possible de passer de la structure arborescente à la structure arc-annotée d'un ARN. Nous nous baserons sur cette représentation pour décrire notre méthode : *RNA-unchained*.

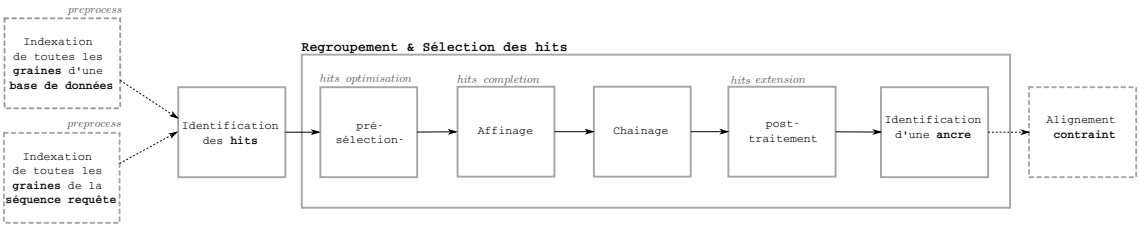


Figure 3.41 – *Architecture du pipeline d'RNA-unchained.*

Comme le montre la Figure 3.41 et comme les filtres BLAST et FastA qui lui sont similaires, RNA-unchained se compose de plusieurs étapes successives :

- Précalcul et indexation de l'ensemble des graines de chacune des séquences cibles T composant le jeu de données D.
- Calcul de l'ensemble des *graines* de la séquence requête Q.
- Identification des graines communes (*hits*) à Q et T.
- (optionnel) Optimisation des hits en fonction de leurs caractéristiques intrinsèques.
- Complétions des hits.
- Chaînages des hits en une *chaîne valide* (*ancre*).
- (optionnel) Extension des hits de l'ancre.

- Alignement contraint par l'ancre (avec *LocARNA*).

Nous nous attacherons dans cette partie à décrire chacune des étapes de RNA-unchained dans leur ordre de traitement par le pipeline.

3.3.1 Modélisation et Indexation des Graines

Afin de mettre en place notre filtre, il convient au préalable de définir le modèle des graines permettant d'établir les hits entre la requête et les cibles. Nous verrons ensuite comment les graines sont indexées de manière à permettre une recherche efficace des graines communes. Enfin nous terminerons cette section par la description d'une optimisation possible des graines basée sur les caractéristiques des structures secondaires d'ARN et améliorant l'alignement final calculé.

(i) Graines (l,d) Centrées

Les graines sont définies comme des motifs en séquences et structures. Comme nous l'avons vu dans la section 1.3.1, même si la séquence présente un certain intérêt, la structure des ARN est tout aussi, voire plus, importante. En effet, cette structure joue un rôle primordiale dans la fonction de l'ARN. Cette structure est d'ailleurs plus souvent conservée entre les ARN d'une même famille, comme par exemple pour les ARNt.

Ces graines sont utiles à deux reprises dans notre programme. Dans un premier temps elles permettent la détection rapide, au sein d'une structure d'indexation notée D, des ARN candidats qui partagent suffisamment de graines communes avec la séquence requête Q. Dans un second temps, un jeu de graines optimales compatibles avec la structure secondaire de Q et des candidats est calculé et sert d'ancre à l'alignement final. La définition de telles graines doit donc :

- permettre une recherche efficace dans la base de données des structures indexées D
- être compatible avec les graines de l'algorithme de chainage utilisé (Allali et al., 2012)

Afin de répondre à ces deux conditions, les graines définies doivent être continues en séquence S et en structure P.

Définition 25 (graines (l, d) centrées)

Soit une séquence arc-annotée $A = (S, P)$ de longueur n et d et l deux entiers tels que $2d \leq l$. Pour tout $i \in \{0, \ldots, n-l\}$, la graine (l, d) centrée de A à la position i, notée cs_i, est la paire (s, p) définie par $p = P[i, i+l]$ et $s = S[i+d, i+l-d]$.

On remarque que dans la définition ci-dessus, s est une séquence de longueur $l - 2d$ sur l'alphabet $\{A, C, G, U\}$ et p est une séquence de longueur l sur l'alphabet $\{., (,)\}$. Ainsi une graine (l, d) centrée n'est plus une séquence arc-annotée puisque

les deux séquences qui la définissent n'ont pas la même longueur et parce que p n'est pas forcément bien parenthésée.

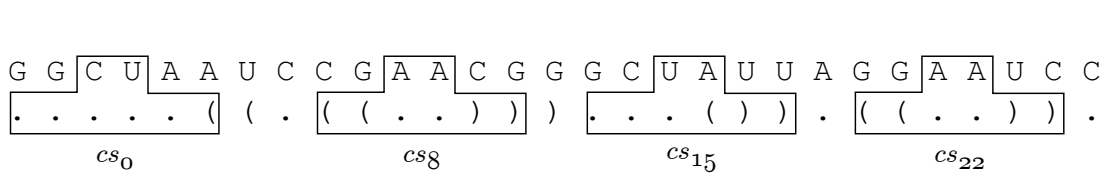


Figure 3.42 – *Exemple de* 4 *graines* $(6, 2)$ *centrées sur une séquence arc-annotée. La première graine* (l, d) *centrée est* cs_0 *et la dernière est* cs_{23}.

Il s'ensuit que, pour deux valeurs l et d, le nombre de graines (l, d) centrées distinctes est $3^l 4^{l-2d}$. De plus, de telles graines peuvent être vues comme des « spaced seeds » (Brown, 2008) sans mismatch en structure mais avec des mismatch possibles en séquence au niveau du préfixe ou du suffixe de taille d de la graine.

Une fois la notion de graine (l, d) centrée définie il faut s'intéresser aux graines communes à deux séquences arc-annotées que nous appellerons *pré hit*.

Définition 26 (pré hit)

Soit $A_1 = (S_1, P_1)$ *et* $A_2 = (S_2, P_2)$ *deux séquences arc-annotées et* d, l *deux entiers tels que* $2d \leq l$. *Un pré hit est une graine* (l, d) *centrée commune à* A_1 *et* A_2, *c'est-à-dire une paire* (i, j) *d'entiers tels que :*

- $0 \leq i \leq |A_1| - l - 1$ *et* $0 \leq j \leq |A_2| - l - 1$,
- $P_1[i, i + l] = P_2[j, j + l]$,
- $S_1[i + d, i + l - d] = S_2[j + d, j + l - d]$.

Le score d'un pré hit S entre deux séquences arc-annotées S_1 et S_2, composé d'une graine (l, d) centrée conservée à la position i sur la séquence S_1 et à la position j sur la séquence S_2, est défini par :

$$score(S) = \sum_{k=0}^{l} f(S_1[i + k], S_2[j + k])$$
$$o\grave{u}\ f(a, a) = 1\ et\ f(a, b) = 0\ si\ a \neq b$$

Il s'en suit que le score s d'un pré hit S vérifie l'inégalité suivante : $l - 2d \leq s \leq l$. Par exemple sur la figure 3.42, le score de la graine $(6, 2)$ centrée "AA";"$((..))$" est 3 et le score de cette seconde graine "AC";"$(..)))$" est 6.

La définition (i) ci-dessus implique que les graines sur les séquences arc-annotées comme définies dans la définition (i) sont des graines valides sur les arborescences ordonnées comme définies dans la définition 3 de (Allali et al., 2012). Les pré hits sont donc compatibles avec l'algorithme de chaînage utilisé (voir la section (i)).

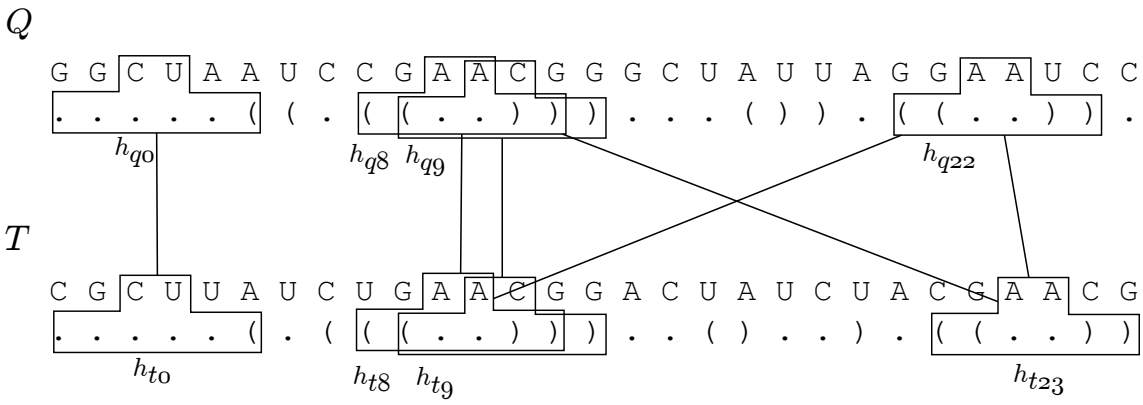


FIGURE 3.43 – *Exemple de* pré hits *entre deux ARN, un ARN requête Q et un ARN cible T. On note que les graines h_{q8} et h_{q9} sur la séquence requête et les graines h_{t8} et h_{t9} sur la séquence cible sont des hits chevauchants. On remarque également que les graines h_{q8} et h_{q22} sur la séquence requête et les graines h_{t8} et h_{t23} sur la séquence cible sont des pré hits qui s'entre-croisent. Ainsi on compte six pré hits au total : $\{(h_{q0}; h_{t0}), (h_{q8}; h_{t8}), (h_{q8}; h_{t23}), (h_{q22}; h_{t8}), (h_{q9}; h_{t9}), (h_{q22}; h_{t23})\}$.*

(ii) Indexation des Graines

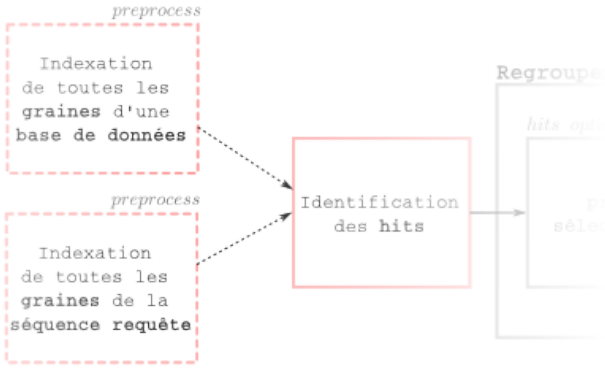


FIGURE 3.44 – *Architecture du pipeline d'RNA-unchained-Étape d'indexation.*

Le premier élément clef de la méthode développée réside dans l'indexation. En effet, pour des paramètres l et d donnés, toutes les graines (l, d) centrées présentes dans les ARN de la base de données d'intérêt à analyser, ou jeu D d'ARN cibles, sont indexées dans une table de hachage. On note $\mathcal{I}_l^d$ cet index. Ainsi pour comparer un ARN Q avec l'ensemble des ARN de D, on commence par rechercher dans $\mathcal{I}_l^d$ les ARN de D qui présentent des graines communes avec Q.

Indexation des graines. Si l'on considère une séquence arc-annotée $A = (S, P)$ de taille n et l, d les entiers paramètres des graines (l, d) centrées. Les $k = n - l + 1$ graines (l, d) centrées issues de A sont indexées dans $\mathcal{I}_l^d$. Pour réaliser cette

indexation toutes les graines (l, d) centrées calculées sont converties en entiers selon le principe suivant : la graine centrée encodant la graine (l, d) centrée à la $i^{\text{ième}}$ position sur la séquence A est définie par

$$\begin{aligned} S_{Value}(A, i, l, d) &= 4^{l-2d} \times \sum_{j=i}^{i+l-1} (encode(P_j) \times 3^{i+l-1-j}) \\ &+ \sum_{j=i+d}^{i+l-d-1} (encode(S_j) \times 4^{i+l-d-1-j}) \\ avec \quad & l'encodage : \ A = 0; \ C = 1; \ G = 2; \ U = 3; \ . = 0; \ (= 1; \ ) = 2 \end{aligned}$$

Ainsi chaque graine a un encodage spécifique. Si l'on considère un entier x, $\mathcal{I}_l^d[x]$ contiendra toutes les occurrences des graines centrées dont la S_{Value} est x, soit :

$$\mathcal{I}_l^d[x] = \{(A, i) \mid S_{Value}(A, i, l, d) = x\}$$

Si l'on prend l'exemple de la séquence arc-annotée de la Figure 3.43, la valeur associée à la graine (l, d) centrée $\{"AA", "((..))"\}$ est $[1\times3^5+1\times3^4+0\times3^3+0\times3^2+2\times3^1+2\times3^0]\times4^2+[0\times4^1+0\times4^0] = 5312$ et $\mathcal{I}_6^2[5312] = \{\{(Q, 8), (Q, 22)\}, \{(T, 8), (T, 23)\}\}$.

On remarque que l'on peut calculer $S_{Value}(A, i+1, l, d)$ à partir de $S_{Value}(A, i, l, d)$ en temps constant. Aussi le calcul des valeurs correspondantes aux graines de A se fait en temps linéaire en la longueur de A. D'un point de vue pratique, un index est créé pour chaque type de graine centrée soit pour chaque couple de paramètres (l, d). Cet index peut être facilement modifié en ajoutant les graines d'une nouvelle séquence ARN. Autant d'index que de couples de paramètres (l, d) voulus peuvent être calculés. Notre pipeline autorise l'utilisation simultanée de plusieurs index pour différentes combinaisons de valeurs pour les paramètres l et d.

Recherche dans l'index Soit une séquence requête Q, la recherche pour aligner cet ARN avec les ARN cibles de la base de données D débute avec le calcul de toutes les graines centrées possibles de Q pour un couple de paramètres (l, d) donnés. Par la suite on recherche l'ensemble $\mathcal{I}_l^d$ des graines de paramètres (l, d) présentes chez tous les ARN de la base D. Pour un ARN donné T de la base de données D, on note $\mathcal{LU}_l^d$ l'ensemble de toutes les graines (l, d) centrées communes à Q et T, soit tous les pré hits entre Q et T :

$$\mathcal{LU}_l^d(Q, T) = \{(i, j) \mid S_{Value}(Q, i, l, d) = S_{Value}(T, j, l, d)\}$$

Par exemple, si l'on reprend la séquence arc-annotée de la Figure 3.43

$$\mathcal{LU}_6^2(A_1, A_2) = \{(0, 0), (8, 8), (8, 23), (22, 8), (9, 9), (22, 23)\}$$

Cette étape est réalisée à l'aide d'une table de hachage standard en utilisant la valeur associée à chaque graine comme clef. Le temps de calcul nécessaire à l'établissement

de $\mathcal{LU}_l^d$ est linéaire, proportionnel à la taille de la séquence en entrée et au nombre total de pré hits. Seules les séquences cibles de D les plus similaires à la séquence requête Q sont nécessaires, c'est pourquoi notre pipeline *RNA-unchained* propose une option permettant de réduire le jeu des candidats à aligner avec Q selon le nombre de pré hits identifiés ou selon le score cumulé de ces pré hits.

(iii) Optimisation des Graines et des Pré Hits [*option*]

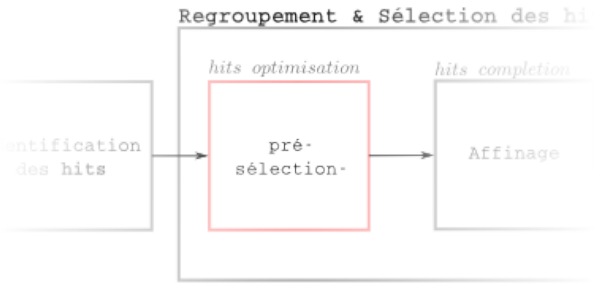


FIGURE 3.45 – *Architecture du pipeline d'RNA-unchained-Étape d'optimisation des pré hits.*

Les expériences préliminaires ont montré que les pré hits ne contenant qu'un seul type d'élément structural, c'est-à-dire dont la séquence P décrivant les éléments structuraux de la séquence ARN ne présentent qu'un seul symbole, étaient souvent de faux signaux positifs (sur la base de l'analyse des alignements de référence de la base de données BraliBase2.1). Dans l'optique d'obtenir des graines plus stringentes, les pré hits calculés dans $\mathcal{LU}_l^d(Q,T)$ ont été filtrés afin de ne conserver que les pré hits présentant au moins deux types de symboles structuraux distincts (soit au moins une paire de bases appariées). Ces pré hits correspondent alors à des éléments structuraux comme les motifs *tiges-boucles :* ( ) et les motifs de *jonctions de tiges :*) (, qui sont des motifs structuraux très conservés et important dans la détection de structures secondaires similaires (Allali and Sagot, 2008; Allali et al., 2008).

Notre pipeline peut prendre en compte cette optimisation en favorisant la conservation de tels pré hits s'ils sont présents dans l'analyse grâce à deux options. Par défaut, *RNA-unchained* conserve l'ensemble des hits identifiés. Une première option permet de ne conserver que les pré hits présentant au moins deux types d'éléments structuraux (option r) tandis qu'une deuxième option permet, s'ils existent, de conserver uniquement les pré hits présentant deux types d'éléments structuraux et, le cas échéant, de conserver tous les pré hits (option rfb).

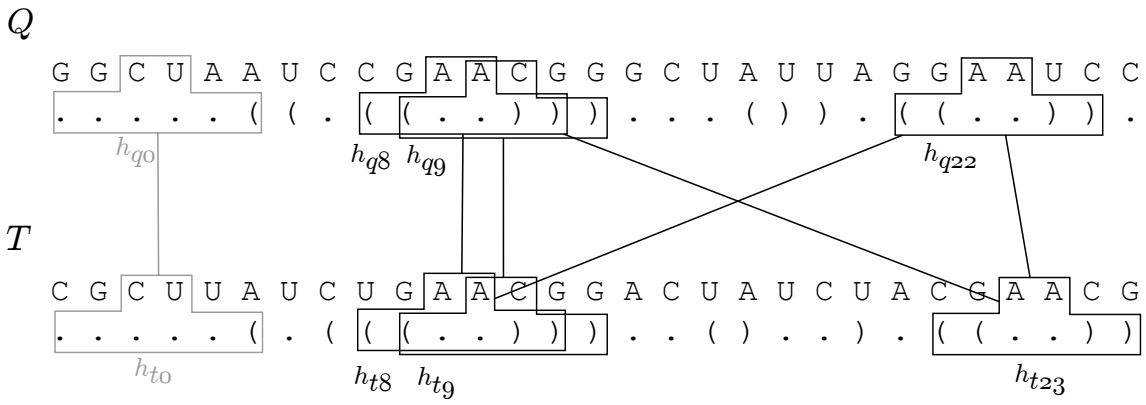


Figure 3.46 – *Par comparaison avec la Figure 3.43, l'un des pré hits est perdu à cause de sa composition structurale. En effet, le hit (h_{q0},h_{t0}) ne se compose pas de deux types de symboles structuraux distincts.*

3.3.2 Recherche de Similitudes de Structures Secondaires d'ARN

Le coeur de l'approche développée ici sur l'alignement d'une séquence requête Q avec un jeu de séquences cibles D repose sur la comparaison au préalable de Q avec chaque membre T de D en se basant uniquement sur leurs pré hits. Pour cela, nous utilisons l'algorithme de chaînage efficace de graines développé dans (Allali et al., 2012) suivi d'une étape au cours de laquelle les gaps entre chaque hits sont comblés par l'algorithme de *LocARNA* (Will et al., 2007).

(i) Complétion des Pré Hits et Algorithme de Chaînage

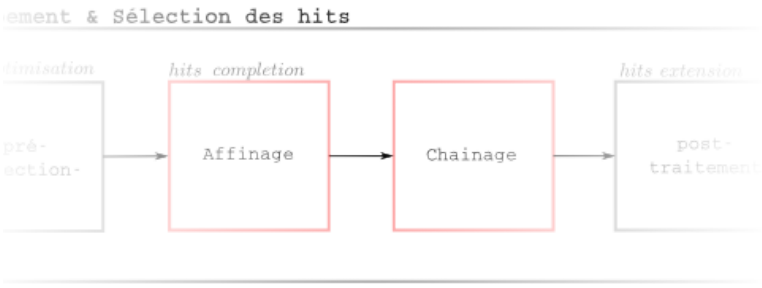


Figure 3.47 – *Architecture du pipeline d'RNA-unchained-Étape de complétion des hits*

La première étape consiste à étendre les graines définissant les pré hits pour tenir compte des appariements donnés par la structure secondaire : Si l'une des graines d'un pré hit ne contient que l'une des deux bases d'une paire, alors la seconde base est rattachée au pré hit, on parle alors de hit. Par exemple sur la Figure 3.48,

la dernière parenthèse de h_{q9} engendre l'extension de la graine avec la parenthèse ouvrante correspondante de Q soit la position 6 (lorsque la première position est désignée par la coordonnée 0). On obtient alors un jeu de hits étendus pour lequel il faut noter que les hits peuvent ne plus être continus sur les séquences comme l'illustre la Figure 3.48.

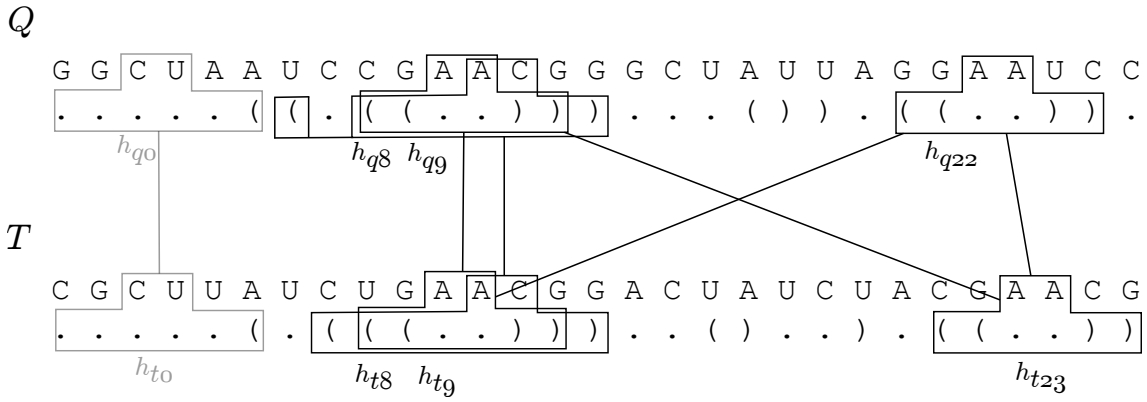


FIGURE 3.48 – *Complétion du pré hit défini par* (h_{q9}, h_{t9}).

Cette étape d'extension revient à exprimer nos pré hits, définis sur des séquences arc-annotées, sur les représentations arborescentes duales de ces séquences. De ce fait, nous allons montrer que ces hits satisfont la définition des hits (mise en correspondance de deux forets internes) donnée par (Allali et al., 2012) et décrite dans la section (i). Il sera alors possible d'utiliser l'algorithme de chaînage rapide sur nos hits.

Propriété 1

Soit f, un pré hit correspondant à un facteur d'une structure parenthésée $A = (S, P)$, de longueur n, défini par un intervalle $[i, j]$, tel que $i \leq j$.
Soit T l'arbre dual de A et la fonction surjective $\alpha : [0, n[\rightarrow T$ qui permet d'obtenir le noeud correspondant à une position de A.

$$f_T = \{f_1 \dots f_k\} \text{ tel que } f_i \in f_T \text{ si } \exists\, x \in [i, j] \text{ tel que } \alpha(x) = f_i$$

Alors f_T est une forêt interne de T telle que définie en (i)

Preuve [*Propriété 1*]

Préliminaires :

Pour tout noeud interne $u \in T$, on définit $\beta_o(u)$ comme l'indice sur S du nucléotide ouvrant la liaison correspondant à u. De même, $\beta_f(u)$ correspond à l'indice du nucléotide fermant cette liaison. Ainsi, si u est un noeud interne, nous avons $\beta_o(u) < \beta_f(u)$. Par extension, si u est une feuille alors $\beta_o(u) = \beta_f(u)$ et correspond à l'indice sur S du nucléotide libre associé à cette feuille.

Soit v un descendant de u dans T, alors pour tout noeud intermédiaire w descendant de u et ancêtre de v, nous avons :

$$\beta_o(u) < \beta_o(w) < \beta_o(v) \leq \beta_f(v) < \beta_f(w) < \beta_f(u)$$

Preuve par l'absurde :

Supposons que f_T n'est pas une forêt interne de T, alors il existe u et v tels que :

(i) soit u est ancêtre de v et il existe w descendant de u et ancêtre de v tel que $w \notin f_T$.
- *u est une liaison (car u a un fils) donc $\beta_o(u) < \beta_f(u)$ et $\alpha(\beta_o(u)) = \alpha(\beta_f(u))$. Par conséquent, $\beta_o(u) \in [i,j]$ ou $\beta_f(u) \in [i,j]$ ou $\beta_o(u), \beta_f(u) \in [i,j]$.*
- *$v \in f_T$. Soit v est une feuille, alors $\beta_o(v) = \beta_f(v) = x$ et $x \in [i,j]$. Soit v est un noeud interne, alors $\beta_o(v) < \beta_f(v)$ et donc $\beta_o(v) \in [i,j]$ ou $\beta_f(v) \in [i,j]$ ou $\beta_o(v), \beta_f(v) \in [i,j]$. Dans ce cas, x est le plus petit indice parmi $\beta_o(v)$ et $\beta_f(v)$ qui appartient à $[i,j]$.*

Si $\beta_o(u) \in [i,j]$, alors $\beta_o(u) < x$ et $]\beta_o(u), x[\subseteq [i,j]$. Or pour tout sommet w entre u et v, w est un noeud interne et $\beta_o(w) \in]\beta_o(u), x[$ (cf. préliminaires) donc $w \in f_T$. De même, si $\beta_f(u) \in [i,j]$.

(ii) Soit il n'existe pas de relation de parenté entre u et v et on suppose, sans perte de généralité, que u est à gauche de v.

Soit u' le plus grand ancêtre de u appartenant à f_T et soit v' le plus grand ancêtre de v appartenant à f_T, sans perte de généralité, on suppose que la hauteur de u' dans T est inférieure ou égale à la hauteur de v'.
- *Soit u' n'est pas frère de v'. Soit alors w ancêtre de v' et frère de u'. Par construction, $\beta_f(u') \in [i,j]$ et $\beta_o(v') \in [i,j]$. Alors nécessairement, $\beta_o(w) \in]\beta_f(u'), \beta_o(v')[$ et donc $w \in f_T$.*
- *Soit u' et v' sont frères mais il existe $w \notin f_T$ frère droit de u' et frère gauche de v'. Comme précédemment on prouve que $\beta_f(u') \in [i,j]$, $\beta_o(v') \in [i,j]$ et $\beta_o(w) \in]\beta_f(u'), \beta_o(v')[$, alors nécessairement $w \in f_T$.*

Nous avons montré que pour tout $u, v \in f_T$, soit ils font partie d'un même arbre interne de T défini par f_T, soit ils appartiennent à deux arbres internes qui composent la forêt interne définie par f_T.

Ainsi, l'étape suivant l'extension de nos pré hits en hits consiste à calculer un sous ensemble de hits compatibles avec les structures secondaires, c'est-à-dire à calculer la chaîne de hits de score maximal.

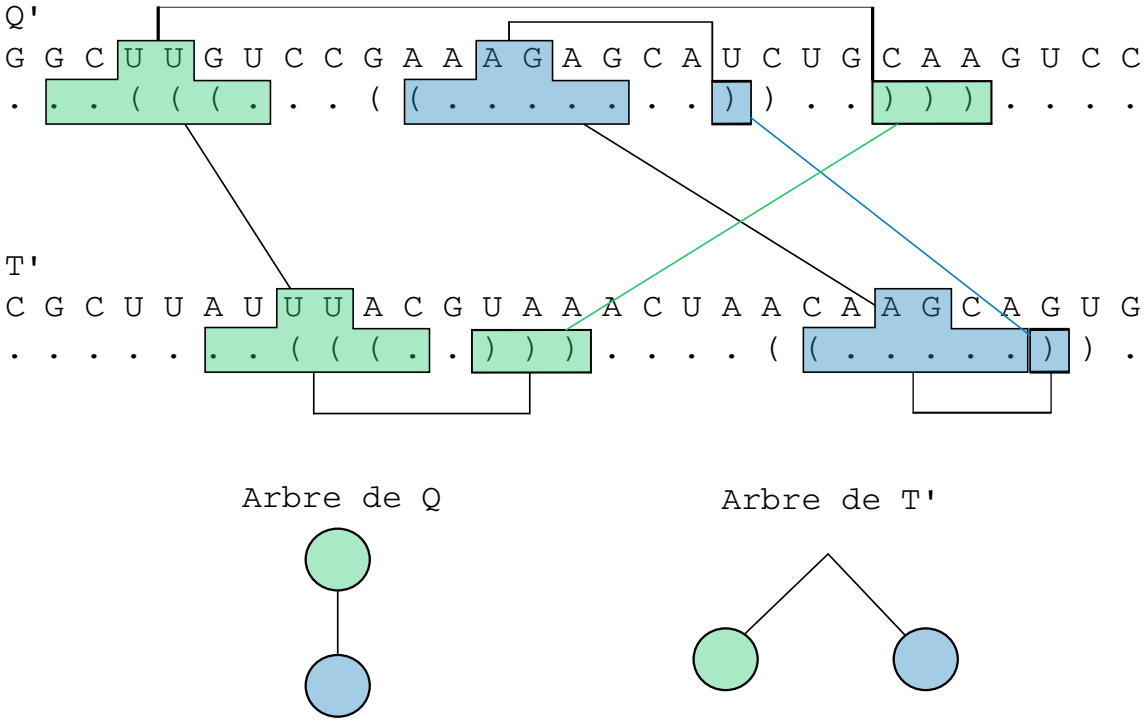


FIGURE 3.49 – *Des hits compatibles doivent préserver les notions d'ancestralité et d'ordre, ce qui n'est pas le cas dans cet exemple comme l'illustre la représentation arborescente des structures.*

Par la suite, on appellera *ancre*, cette chaîne de hits projetée sur les séquences arc-annotées. Ainsi, une ancre se définie par un ensemble de triplets $\{(i, j, r)\}$, chaque triplet (i, j, r) indiquant que les nucléotides $i \dots i+r$ de Q sont mis en correspondance avec les nucléotides $j \dots j + r$ de T. Par construction, pour tout couple de triplet (i, j, r), (i', j', r') :

- si $i = i'$ alors $j = j'$ et $r = r'$
- si $i < i'$ alors $i' > i + r$ et $j' > j + r$
- si $i > i'$ alors $i > i' + r'$ et $j > j' + r'$

Ainsi, une ancre correspond à un ensemble de mises en correspondances entre Q et T de facteurs colinéaires et non chevauchants [7]. Dans ce cas on parle alors de facteurs *compatibles*.

La notion de colinéarité dans les structures secondaires correspond à la préservation des relations d'ancestralité et d'ordre dans la représentation arborescente des structures secondaires (voir Figures 3.49 et 3.50) comme introduit par (Jiang et al.,

7. Une définition plus précise est présentée dans (Allali et al., 2012) pour les ancres sous le nom de *chaînes*

1995). Le score d'une ancre C est donné par la somme des scores des graines définissant les hits qu'elle contient. Le score de chaînage entre deux séquences S_1 et S_2 est le score maximal parmi tous les scores des ancres calculées, on parle alors d'ancre optimale. L'algorithme utilisé ici calcule l'ancre optimale en $O(k^2 \log k)$ en temps, où k est le nombre de hits. Si l'on reprend le même exemple que précédemment, une ancre est composée des hits $\{\{(Q,8),(T,8)\},\{(Q,22),(T,23)\}\}$ (voir la Figure 3.46). Après l'étape de chaînage, nous avons pour chaque ARN T de D un ensemble de hits entre Q et T qui forment une ancre optimale $\mathcal{A}(Q,T)$. On appelle *gaps* les segments des ARN Q et T composés de nucléotides qui n'appartiennent pas aux hits de l'ancre.

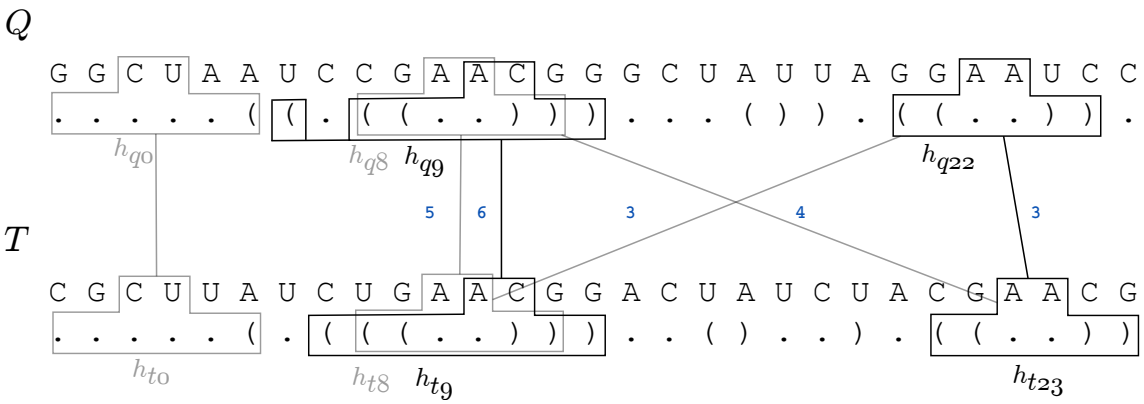


Figure 3.50 – *Les pré hits chevauchants, (h_{q8},h_{t8}), (h_{q9},h_{t9}), ou qui s'entrecroisent, (h_{q8},h_{t23}) et (h_{q22},h_{t8}), sont écartés au cours de l'étape de chaînage (en fonction de leur score). Ainsi cet exemple présente les hits finaux entre les séquences Q et T.*

(ii) Extension des Ancres [*option*]

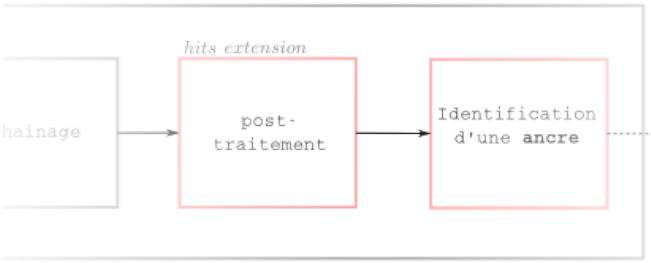


Figure 3.51 – *Architecture du pipeline d'RNA-unchained-Étape de complétion des hits.*

Avant d'aligner les gaps des séquences Q et T avec un algorithme d'alignement exact mais plutôt coûteux, une dernière phase optionnelle d'extension des graines permet de réduire la taille des gaps (option *epc*). En outre, cette étape d'extension permet de pallier la taille limitée des graines initiales. Dans un premier temps, chaque hit entre les deux ARN Q et T est de nouveau étendu de part et d'autre de chaque graine qui le compose s'il y a une similarité exacte en séquence. Cette nouvelle étape est décrite dans l'exemple présenté dans la Figure 3.52 où le hit (h_{q22}, h_{t23}) est étendu à gauche de deux nucléotides.

Puis on applique sur les zones de gaps restantes un algorithme adapté du LCS (*cf.* (i)). Sur deux séquences l'algorithme LCS consiste à identifier la plus longue chaîne de caractères telle que chaque lettre de ce mot apparaît dans le même ordre dans chaque séquence. Afin d'éviter l'ajout de contraintes peu significatives, l'algorithme LCS a été calculé sur des triplets de caractères (voir la Figure 3.52 pour un exemple). Pour cela chacune des séquences a été encodée à l'aide d'une fenêtre glissante de trois caractères avec un pas de un. Ainsi la séquence est encodée par une suite de nombre codant pour un des triplets successifs de la séquence. L'algorithme de LCS est alors appliqué à ces séquences ce qui assure la condition d'au moins trois nucléotides successifs similaires.

Cependant cette amélioration de la contrainte n'est significative qu'à condition que la couverture de l'ancre soit suffisamment élevée. Pour cela un seuil est utilisé. Pour appliquer l'optimisation à l'aide du LCS, l'ancre doit avoir une taille au préalable supérieur à $4 \times l$ bases.

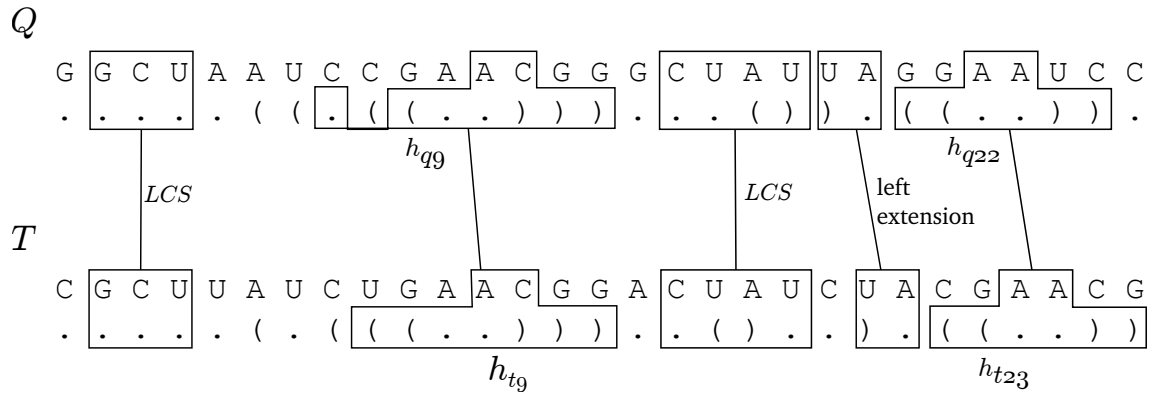


FIGURE *3.52 – L'ancre entre Q et T est étendue pour le hit h_{q23}-h_{t23} à gauche. Puis le calcul de la LCS sur les gaps à permis de les combler en partie.*

(iii) Alignement des Structures Contraintes avec *LocARNA*

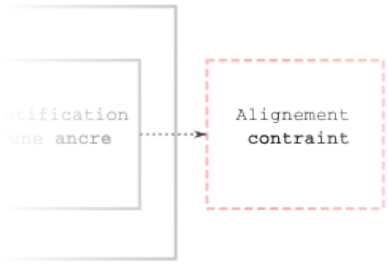


Figure 3.53 – *Architecture du pipeline d'RNA-unchained-Étape d'alignement final contraint.*

Pour terminer, pour chaque candidat T homologue à la séquence Q, les gaps définis par l'ancre calculée entre Q et T sont alignés grâce à l'algorithme *LocARNA* (Will et al., 2007). Pour cela *LocARNA* est utilisé avec l'option de contraintes qui permet de fournir l'ancre calculée comme une suite de contraintes à respecter (voir la Figure 3.54).

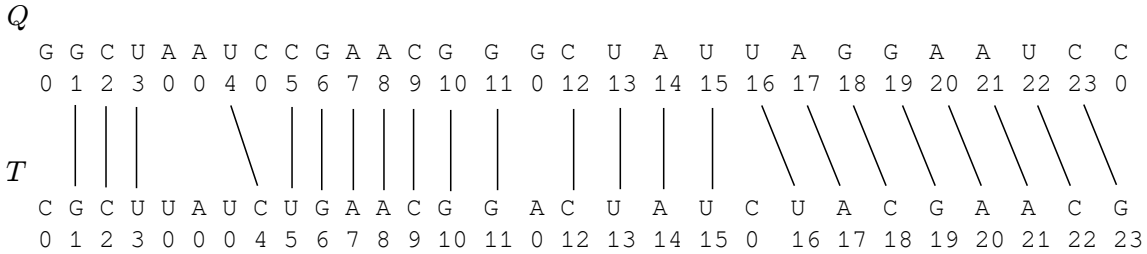


Figure 3.54 – *L'ancre calculée pour notre exemple vue comme une séquence de contraintes.*

3.4 Tests et Performance du Filtre

Afin d'analyser les capacités du pipeline implémenté à produire des alignements corrects, nous avons utilisé un jeu de données d'étalonnage, le benchmark BraLi-Base2.1 (Wilm et al., 2006), qui se compose d'un jeu de séquences ARN alignées. Nous avons donc calculé à nouveau les alignements de ce benchmark à partir des séquences fournies. Pour cela nous avons utilisé notre pipeline *RNA-unchained* avec différentes combinaisons d'options afin d'en analyser les impacts et de peut être pouvoir en déduire une combinaison optimale. Nous comparerons alors les résultats

obtenus par *RNA-unchained* avec un programme d'alignement reconnu, *LocARNA*, et parmi les plus performant en terme de qualité et de temps de calcul.

3.4.1 Outils d'Analyse et de Comparaison

Afin de pouvoir tester les différents paramètres de *RNA-unchained* mais également de comparer la qualité de ses résultats avec d'autres outils actuels de comparaison de séquences, il est important de déterminer un jeu de données calibré mais aussi des valeurs statistiques d'analyse appropriées.

(i) BraliBase comme Base de Données de Test

BraLiBase2.1 (Wilm et al., 2006) est un benchmark composé de 8 976 alignements composés de deux séquences. Ces alignements sont eux même classés par familles. BraLiBase2.1 compte ainsi 36 familles d'ARN différentes. Dans chaque famille on décompte un nombre variable de séquences d'ARN mais tous les ARN d'une même famille ne sont pas alignés les uns avec les autres. Dans une même famille un même ARN n'est aligné que sept fois au maximum. BraliBase2.1 contient ainsi 7 858 séquences différentes. Ce benchmark a la particularité de présenter les alignements de référence pour chaque couple d'ARN qu'il contient. Ces alignements ont été vérifiés et validés manuellement par des biologistes. De plus avec BraLiBase2.1 il est possible de récupérer en parallèle le script *compalignp* qui permet de comparer les alignements obtenus avec les alignements de référence de BraLiBase2.1 en se basant sur le *SPS* (*Sum of Pair Score*).

(ii) Valeurs d'Analyse de la Qualité d'Alignement

Sum of Pair Score Afin de comparer les alignements obtenus aux alignements de référence de BraliBase2.1, nous utiliserons la valeur statistique du SPS (*Sum of Pair Score*) qui a également été utilisée par Schmiedl et al. (2012). Initialement introduit pour évaluer la qualité des alignements de séquences multiples, le SPS est défini comme la proportion de paires de bases correctement alignées dans l'alignement calculé par rapport à l'alignement de référence. Plus précisément, soient un alignement de référence r de taille l_r et un alignement calculé e de taille l_e, le SPS est donné par le ratio $\frac{SP^e}{l_r}$. On appelle SP^e le nombre de paires (i, j) telles que les positions i et j sur les séquences originales sont des mises en correspondances dans chacun des deux alignements r et e. Même si ce score est utilisé fréquemment lors d'études d'alignements multiples de séquences et est fournis dans de nombreux paquetages, il est difficile d'expliquer d'un point de vue biologique la signification de la valeur du SPS.

Couverture En outre du SPS, on prend également en compte le taux de couverture par les ancres des ARN alignés. Ce taux est calculé comme le ratio entre le

nombre de bases appartenant à l'ancre et la longueur de l'alignement (Figure 3.55). Le taux de couverture par les ancres est important car il permet d'évaluer l'impact de ces ancres aussi bien du point de vue du temps de calcul que de la justesse des résultats. En effet, un taux de couverture élevé mais relevant de mauvaises ancres mènera nécessairement à des alignements de mauvaise qualité et donc à une faible valeur de SPS alors qu'un faible taux de couverture mais plus fiable pourra générer un alignement de meilleur qualité mais sans gain significatif du temps de calcul.

Similarité Enfin, afin de pouvoir analyser les valeurs de SPS et de couverture des ancres nous les présentons en fonction de la similarité entre les deux ARN comparés dans leur alignement de référence. On définit cette similarité $Sim(Q,T)$ entre une séquence requête Q et une séquence cible T, deux séquences appartenant à un même alignement de référence, comme :

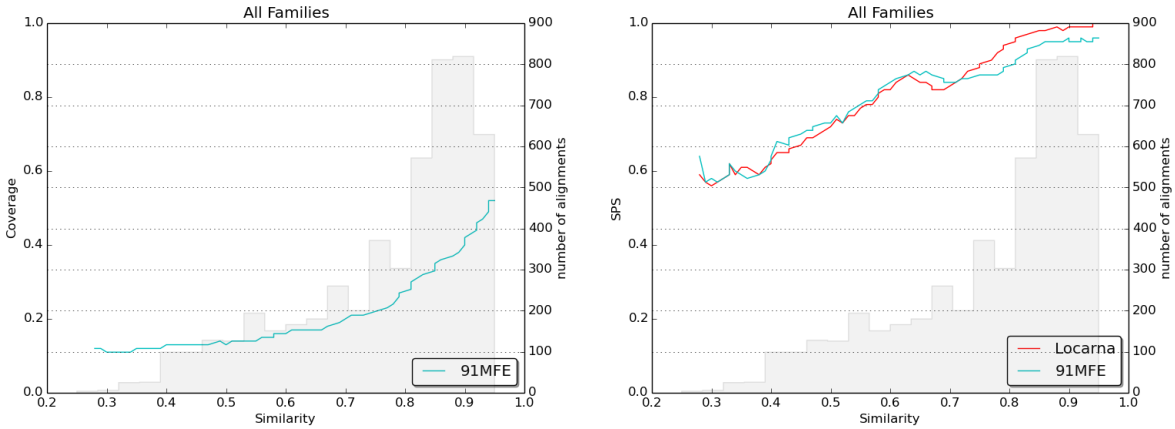
$$Sim(Q,T) \quad = \quad \frac{\sum_{i=0}^{|Q'|-1} f(Q'[i],T'[i])}{|Q'|}$$

où Q', T' font références aux séquences Q et T alignées, et $f(a,a) = 1$ et $f(a,b) = 0$ si $a \neq b$.

Présentation des résultats Les deux Figures 3.55.a et 3.55.b présentent les résultats, respectivement, de SPS et de couverture, obtenus avec *RNA-unchained* (Echelle de gauche). En outre de ces résultats, les histogrammes présentent le nombre d'alignements (échelle de droite) par intervalles de similarité (avec un pas de 5). Dans les deux cas, les courbes sont générées en fonction du taux de similarité des alignements de référence.

Afin de minimiser l'impact sur la courbe de l'utilisation de *LocARNA* sans contraintes tout en conservant suffisament d'alignements pour l'analyse, seuls les alignements présentant au moins un hit suite au chainage avec *RNA-unchained* ou une EPM suite au chainage avec *ExpLocP* sont conservé pour tracer les courbes.

Ainsi si on observe sur la Figure 3.55.a de faibles valeurs de couverture, cela signifie que peu de hits ont été calculés entre les ARN étudiés, ce qui peut ne pas être suffisant pour contraindre suffisamment l'alignement avec *LocARNA* et n'influera pas sur son temps de calcul. Au contraire, une valeur élevée de couverture est synonyme d'un grand nombre de hits calculés pour ces séquences ARN et cette forte couverture pourrait permettre d'améliorer les temps de calcul d'alignement avec *LocARNA*. Cependant un grand nombre de hits peut également provenir de critères trop peu stringents et certains de ces hits pourraient être de faux positifs qui faussent alors l'alignement.
Sur la Figure 3.55.b, on observe alors la qualité des alignements générés par rapport aux alignements de références fournis par BraliBase2.1. Plus cette valeur est élevée plus l'alignement calculé est proche de celui de référence.



(a) Analyse de la couverture des graines

(b) Analyse de la qualité des alignements

FIGURE *3.55 – Exemple de courbes présentant les résultats d'alignement.*

Dans le fond de ces deux figures on observe un histogramme reflétant le nombre d'alignements pris en compte pour le calcul du point de la courbe pour chaque intervalle de similarité en séquence des alignements.

(iii) Outils d'Alignements

Nous avons utilisé l'outil d'alignement exact *LocARNA* (Will et al., 2007) qui sert alors de référence à l'ensemble des études réalisées. En effet, *LocARNA* est un outil qui accepte en entrée deux séquences ARN puis qui calcule pour ces deux ARN la matrice de probabilité d'appariement des résidus. Enfin à partir de cette matrice *LocARNA* extrait l'alignement le plus probable avec une complexité de l'ordre de $O(n^4)$ (voir Section 3.1.2(iv)).

Nous avons calculé ces alignements avec un filtre de *LocARNA*, ExpaRNA (Heyne et al., 2009), qui est un pipeline similaire à *RNA-unchained*. *ExpaRNA* est un filtre permettant d'améliorer les temps de calcul de *LocARNA*. Cependant tout comme *LocARNA*, *ExpaRNA* utilise la matrice de probabilité des ARN qui sont à aligner ; il n'a donc pas besoin de la structure secondaire des ARN. De plus il utilise un modèle de graines appelées EPM qui correspondent aux plus longs motifs de similitude exacte entre les deux séquences. Mais contrairement à *RNA-unchained* qui présente une phase de calcul précédant l'analyse et permettant l'utilisation d'un index, *ExpaRNA* calcule les EPM directement sur les deux séquences qui lui sont fournies en entrée. Ainsi, ces deux outils, *LocARNA* et *ExpaRNA*, calculent l'alignement optimal grâce aux scores de la matrice de probabilités d'appariements des bases, c'est-à-dire à partir de l'ensemble des possibilités de repliement des deux ARN analysés. Ils sont à la pointe de leur domaine. Afin de pouvoir reproduire les analyses réalisées avec BraliBase2.1 par les auteurs dans (Heyne et al., 2009), nous avons obtenu des

auteurs de l'article le code et les paramètres qu'ils avaient alors utilisés. Nous avons ainsi utilisé deux modes pour *ExpaRNA* :

- avec les paramètres optimisés fournis par les auteurs, que l'on appellera *ExpLocP*
- avec les paramètres par défaut, que l'on appellera *ExpLocPNoOpt*.

3.4.2 Analyses Comparatives de l'Impact des Différentes Graines Sélectionnées

(i) Différentes Méthodes de Génération de la Structure Secondaire

Afin d'utiliser la méthode développée dans *RNA-unchained*, les séquences ARN doivent être fournies avec leurs structures secondaires. Cependant, la plupart du temps au cours des expériences seule la séquence des ARN est disponible. En effet, comme discuté au début de ce manuscrit, la structure secondaire dépend de l'environnement du polynucléotides et il est difficile de l'obtenir, aussi bien d'un point de vue technique que financier. De nombreux logiciels de repliement d'ARN ont été développés et permettent à partir de la structure primaire d'accéder à une structure secondaire possible.

La structure *MFE* La structure « Minimum Free Energy (MFE) » d'une séquence ARN est la structure secondaire de cet ARN qui contribue à obtenir l'énergie libre minimale d'appariement pour cette séquence. Une telle structure est prédite à partir d'un modèle d'énergie basé sur les boucles de repliement et d'un algorithme de programmation dynamique introduit par Zuker and Stiegler (1981). En tant que structure secondaire un ARN peut être décomposé en boucles et en bases non appariées. Le modèle d'énergie basé sur les boucles calcule alors l'énergie libre $F(S)$ d'une structure secondaire d'ARN S comme la somme des énergies libres, F_L, de chaque boucle L composant S. Ainsi pour un jeu de paramètres d'énergie et une température (par défaut $37C$) donnés, la structure secondaire S minimisant la valeur de $F(S)$ est calculée comme la somme minimisant cette valeur.

Afin d'obtenir la structure MFE, on utilise le programme *RNAfold* du *ViennaPackage* (Lorenz et al., 2011).

La structure *MEA* La notion de structure secondaire MEA, pour *Maximum Expected Accuracy*, a été introduite par *Do et Al.* (Do et al., 2005). Une fonction de partition basée sur des méthodes thermodynamiques permet d'obtenir les probabilité d'appariement des bases. Les structures identifiées sont alors assemblées grâce à un algorithme de programmation dynamique et la combinaison de structures qui maximise le score final est sélectionnée.

De même que pour calculer la structure MFE nous utilisons *RNAfold* du *ViennaPackage* (Lorenz et al., 2011) pour obtenir la structure MEA des séquences en entrée de RNA-unchained.

Les shapes Les shapes constituent une représentation de la structure d'un ARN. Différents types de shapes (cinq) existent. Le type de shape sélectionné représente le niveau d'abstraction (ou de dissimilarité) pour lequel la shape obtenue est différente du niveau précédent. Globalement, les région hélicoïdales sont représentées par une paire de crochets ouvrant et fermant et les régions non appariées sont assimilées à un unique tiret « _ ». Les différentes shapes sont dues à l'implication ou non d'éléments structuraux tels que les renflements, les boucles internes, les boucles multiples ou encore les tiges, dans la représentation de la shape (voir la Figure 3.56). On distingue ainsi cinq types de shapes :

-Type 1 Le plus précis - Toutes les boucles et toutes les régions non appariées sont symbolisées.

-Type 2 Tous les types de boucles ainsi que les régions non appariées situées dans des boucles externes et des boucles multiples.

-Type 3 Seuls les différents types hélices sont représentés (les régions non appariées sont omises).

-Type 4 Seules les hélices des boucles externes ou des boucles multiples sont représentées.

-Type 5 Le plus abstrait - Seules les tiges (imbriquées) sont prises en compte.

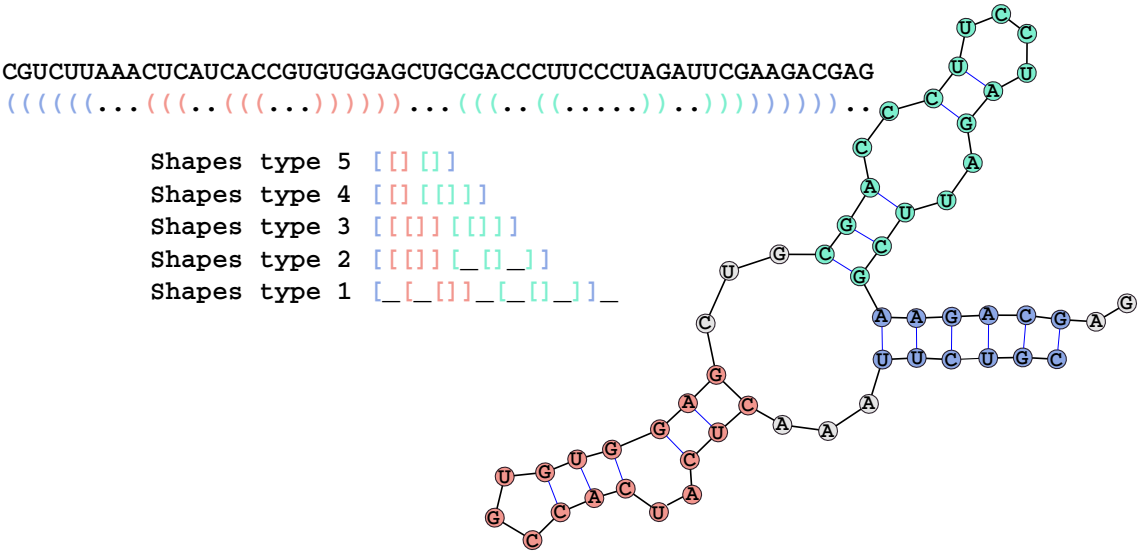


FIGURE *3.56 – Exemple des cinq shapes possibles pour un ARN donné.*

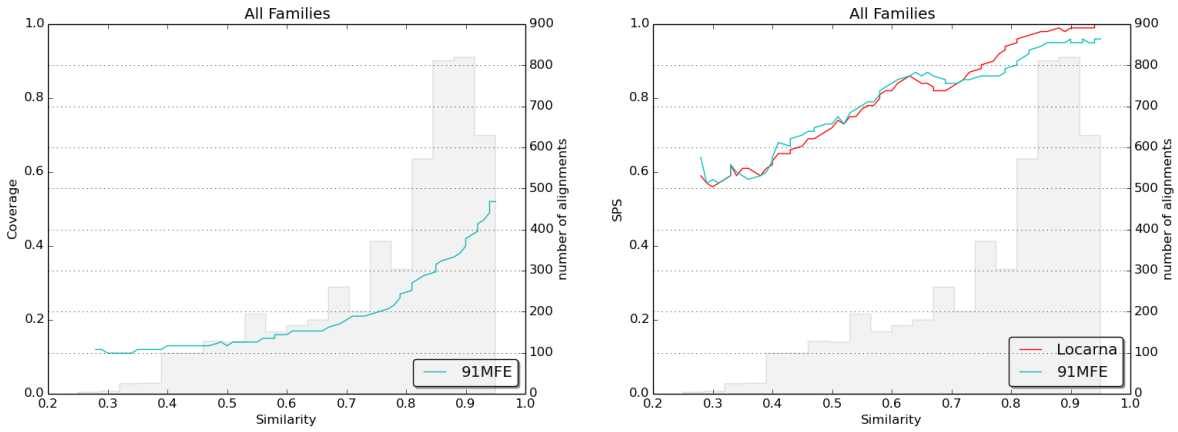
RNAshapes (Steffen et al., 2006) est un programme permettant de calculer pour une séquence donnée en entrée les différentes shapes voulues de cet ARN. Il est alors possible en choisissant un type de shape de récupérer pour un ARN donné la structure secondaire associée à cette shape.

L'approche multistructures La structure la plus stable ou la plus probable n'est pas nécessairement la structure secondaire réelle d'un ARN. C'est pourquoi

nous nous sommes intéressés à une approche multi structures au cours de laquelle non pas une mais k structures seront sélectionnées parmi les plus probables.

Chacune des k structures de la séquence en entrée est comparée aux k structures de chacun des ARN de la base de données. Pour deux ARN comparés on effectue alors $\frac{k^2-k}{2}$ comparaisons et on ne retient que le chainage menant au meilleur alignement.

Cependant cette méthode présente un certain nombre d'inconvénients. Dans un premiers temps on calcul k structures. Dans un second temps $\frac{k^2-k}{2}$ chaînages sont effectués pour un seul couple d'ARN, le temps de calcul est donc démultiplié.



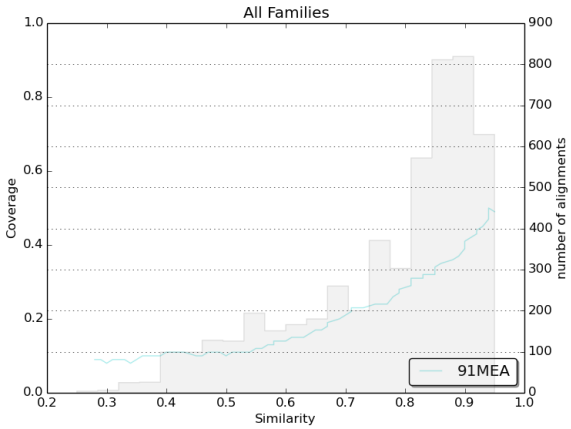
(a) Analyse de la couverture des graines.

(b) Analyse de la qualité des alignements.

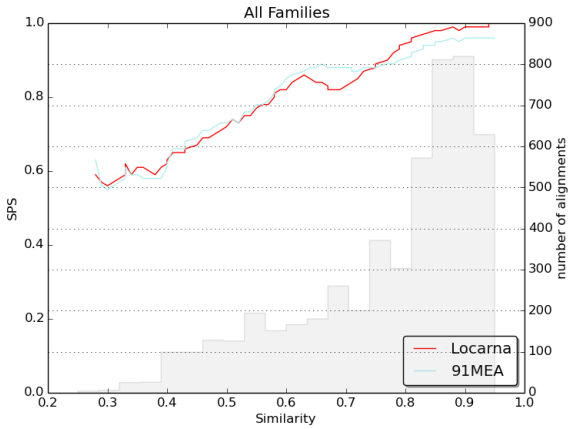
Figure 3.57 – *Impact de la méthode de repliement MFE sur la comparaison d'ARN.*

Impact de la méthode de repliement sur les résultats Chacune de ces méthodes permettant d'obtenir une structure secondaire pour les ARN analysés a été testée (voir Figures 3.57, 3.58, 3.59 et 3.60). On observe que, pour l'approche multistructure, un nombre plus important de structures pour un ARN donné permet d'améliorer la couverture par les graines de l'alignement et la qualité des alignements. Mais les temps de calculs étant multipliés par le nombre de structures considérées, les gains en qualité ne sont pas importants. Il en va de même pour le calcul des shapes qui est très prenant en temps (aussi long que le chaînage lui-même) et qui, de plus, n'améliore pas la qualité des alignements calculés.

Pour le reste de l'étude et suite à ces résultats, nous avons conservé le repliement selon la structure secondaire MFE obtenue avec l'outil *RNAfold*. Nous dénommerons par « MFE » les courbes générées à partir de ce repliement.

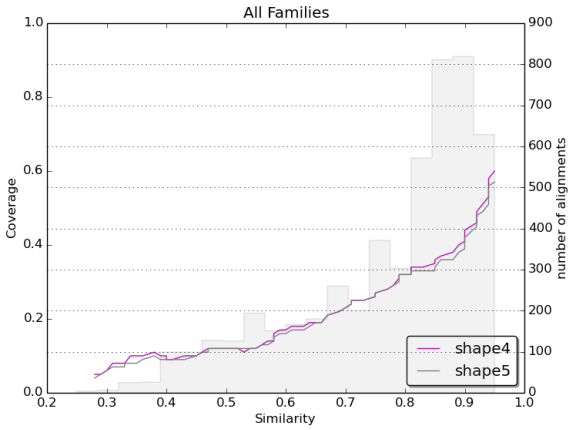


(a) Analyse de la couverture des graines.

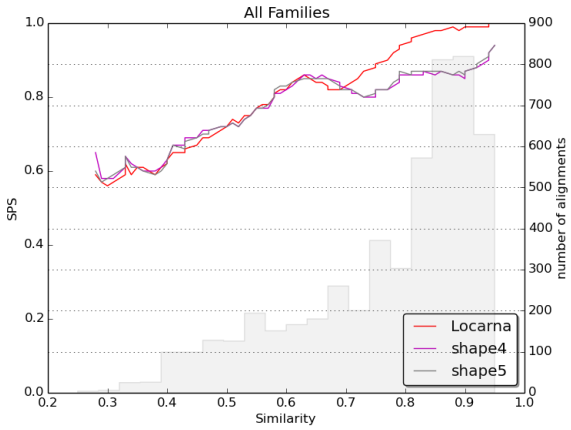


(b) Analyse de la qualité des alignements.

FIGURE *3.58 – Impact de la méthode de repliement MEA sur la comparaison d'ARN.*



(a) Analyse de la couverture des graines.



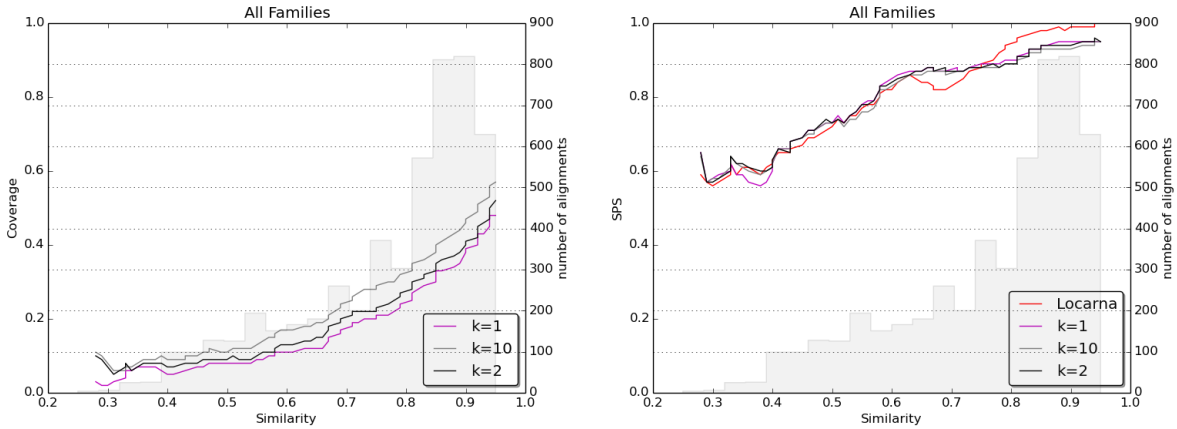
(b) Analyse de la qualité des alignements.

FIGURE *3.59 – Impact de la méthode de repliement des shapes sur la comparaison d'ARN.*

(ii) Différentes Tailles de Graines

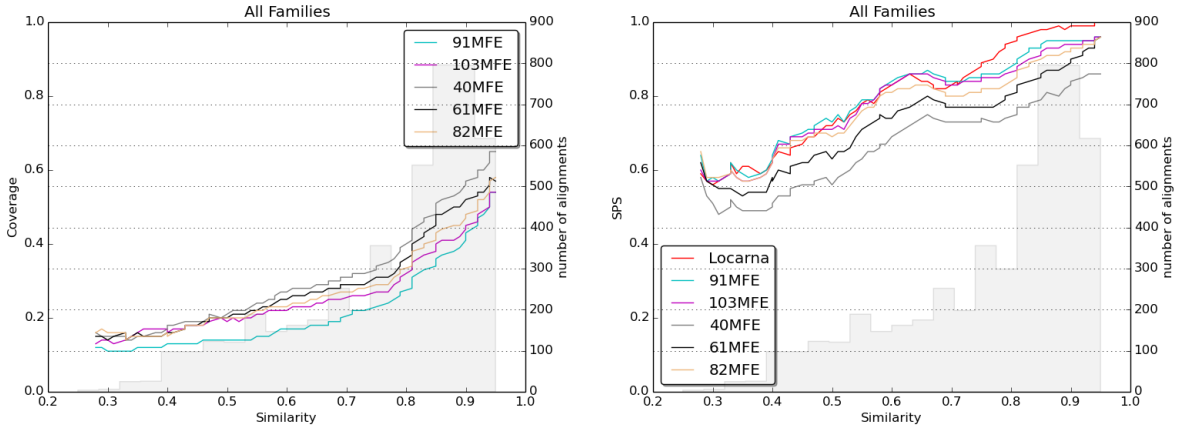
Une graine (l, d) centrée, dépend de ses paramètres l et d. Il est donc intéressant d'évaluer l'impact de différentes valeurs pour l et d. Il est alors important de remarquer que plus la valeur de d est petite plus les graines seront stringeantes puisque on se rapproche des EPM qui sont très proches des graines $(l, 0)$ centrées.

Concernant les paramètres de définition des graines, plusieurs valeurs pour l et d allant de $(4, 0)$ à $(10, 3)$ ont été testées afin de déterminer une taille de graine adaptée permettant d'obtenir une couverture des séquences suffisante pour produire



(a) Analyse de la couverture des graines. *(b) Analyse de la qualité des alignements.*

Figure *3.60 – Impact de la méthode de repliement multi-structures sur la comparaison d'ARN. k représente le nombre de structures secondaires générées pour chaque ARN, soit ici on a généré les courbes pour* 1, 2 *et* 10 *structures secondaires pour chaque ARN.*



(a) Analyse de la couverture des graines *(b) Analyse de la qualité des alignements*

Figure *3.61 – Importance de la taille de la séquence de la graine.*

des alignements de qualité sans pour autant impacter sur les temps de calculs.

Les expériences que nous avons effectuées sur une large gamme de valeurs pour l et d (voir Figures 3.61, 3.62 et Annexe 81) montrent clairement que de petites portions de séquences conservées génèrent de nombreuses graines qui sont de faux marqueurs positifs. Ce qui est par exemple le cas pour les graines $(5, 1)$ (voir Annexe 81) qui impliquent la conservation d'une courte séquence de trois nucléotides consécutifs.

Pour une même taille en séquence, soit pour des valeurs de $l - 2d$ identiques

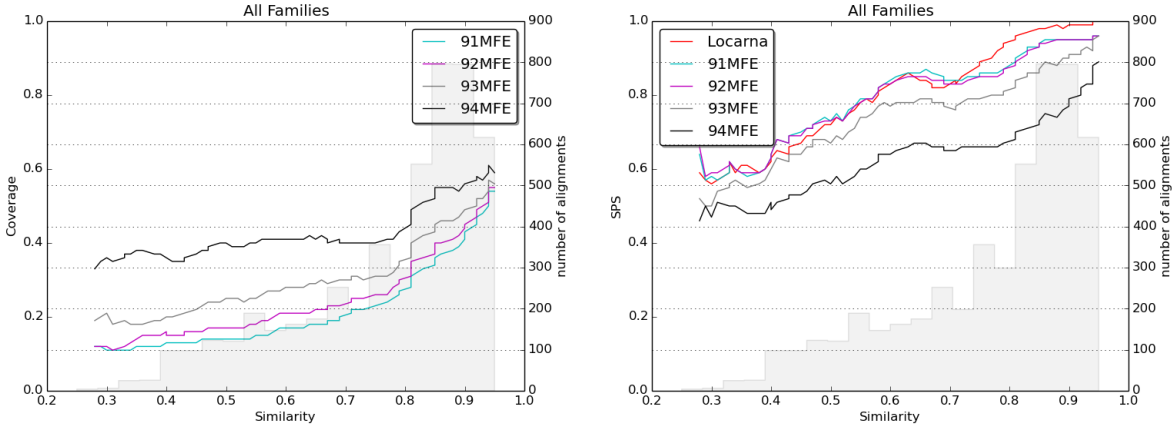
(voir Figure 3.61), une petite taille en structure est à l'origine de nombreux faux positifs et d'alignements de faible qualité. Une taille élevée en structure permet alors de compenser l'effet de la petite taille en séquence et améliore la qualité des alignement. La taille de la structure conservée a ainsi un impact sur la spécificité des graines sélectionnées.

Une étude similaire portant sur l'implication de la taille de la structure dans la qualité des résultats d'alignements est réalisée.

Au contraire, la conservation d'une longue portion de structure dans les graines génère un faible taux de couverture des séquences ARN par ces graines. Ceci se vérifie d'autant plus que l'on applique l'option de filtrage sur la composition en structure des graines, c'est-à-dire que toute structure doit présenter au moins deux types de caractères structuraux (voir la discussion plus loin).

De manière plus générale, on observe une corrélation entre un taux de couverture par les graines élevé et une faible valeur du SPS, soit des alignements de moindre qualité.

Enfin les graines dont les paramètres ont des valeurs proches, comme par exemple $(9, 1)$, $(9, 2)$, $(8, 1)$ et $(8, 2)$ présentent des performances relativement similaires.

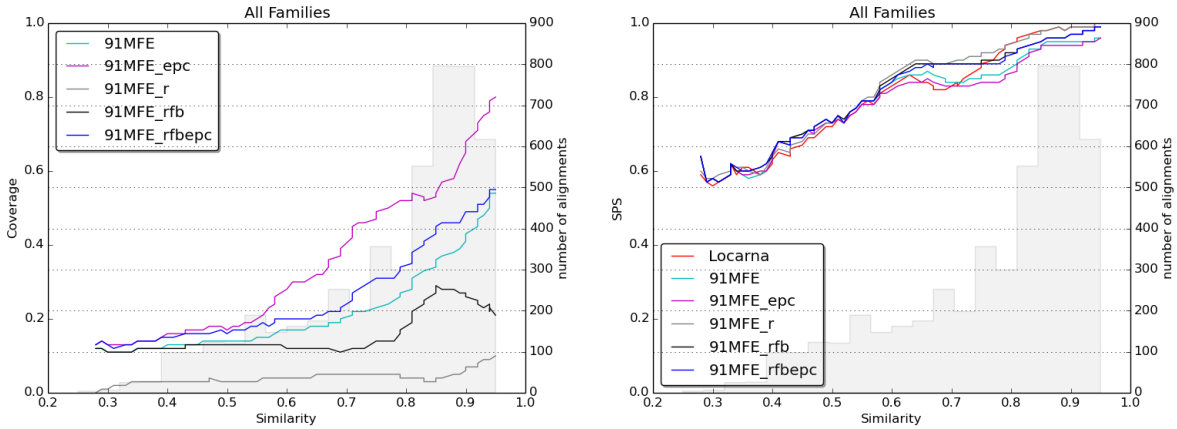


(a) Analyse de la couverture des graines

(b) Analyse de la qualité des alignements

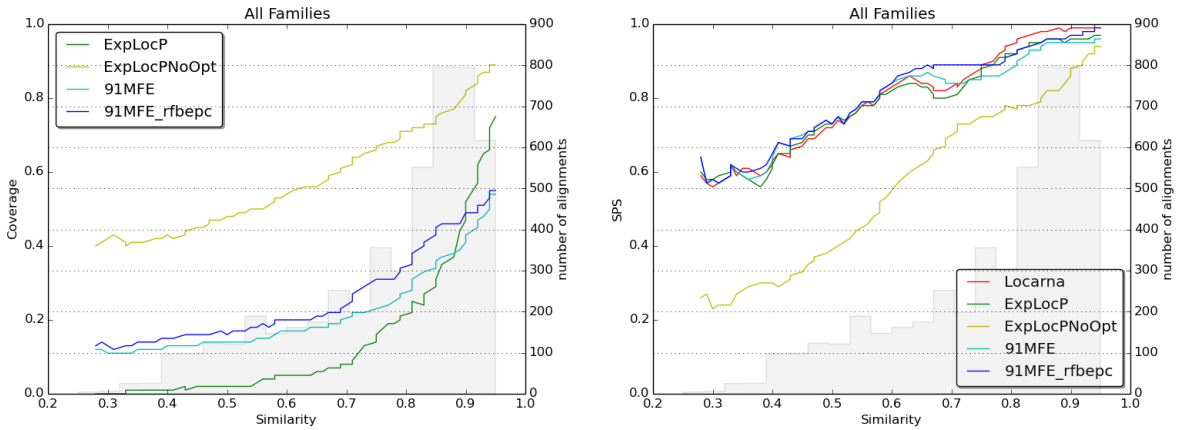
FIGURE *3.62 – Importance de la taille de la structure de la graine.*

Pour la suite des analyses nous avons sélectionné les graines de paramètres $(9, 1)$ qui présentent des alignements de bonne qualité et une couverture correcte des graines permettant d'améliorer les temps de calcul des alignements finaux. Nous dénommerons par « 91MFE » la courbe par défaut de *RNA-unchained* (c'est-à-dire sans options r, rfb et epc).



(a) Analyse de la couverture des graines *(b) Analyse de la qualité des alignements*

Figure 3.63 – *Impact des options d'*RNA-unchained *sur la comparaison d'ARN.*



(a) Analyse de la couverture des graines *(b) Analyse de la qualité des alignements*

Figure 3.64 – *Comparaison des résultats obtenus avec les outils* RNA-unchained, LocARNA *et d'*ExpLocP, *avec et sans option.*

(iii) Différentes Options d'Optimisation des Graines

Dans l'optique d'évaluer l'impact de l'utilisation de nos options sur les alignements finaux nous avons utilisé les graines de paramètres $(9, 1)$. Chacun des critères de sélection des graines à chainer décrits dans le paragraphe (iii) est testé indépendament (voir Figure 3.63) :

- courbe $91MFE\_r$: cette courbe permet d'analyser l'impact de la sélection stringente des seuls hits présentant deux types de structures. Si ce n'est pas le cas, l'alignement est obtenu en utilisant *LocARNA* (Will et al., 2007) sans

contrainte.

- courbe $91MFE\_rfb$: ici on conserve les hits présentant au moins deux structures différentes mais si aucun des hits ne présente cette caractéristique on conserve l'ensemble des hits initiaux.

De même, l'impact de l'extension des ancres après chaînage (ii) a également été analysé.

- courbe $91MFE\_epc$: dans ce cas les graines chaînées sont étendues selon les critères discutés auparavant.

Enfin nous avons combiné les deux types de critères, rfb et *epc*, dans la courbe $91MFE\_rfbepc$.

Nous avons comparé les résultats des combinaisons de nos options avec trois programmes de référence :

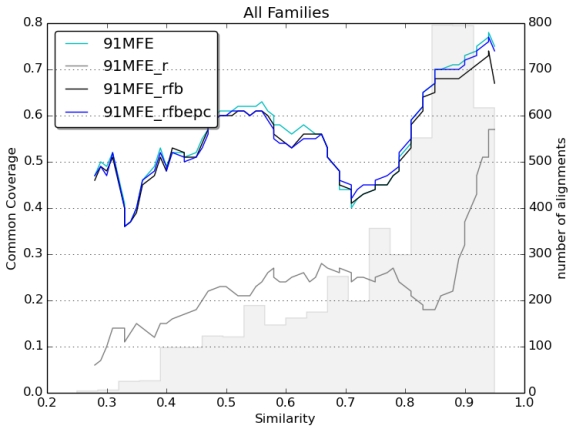
- *LocARNA* : en rouge, la courbe présentant les résultats d'alignement avec *LocARNA* seul.
- *ExpLocP* : en vert, la courbe présentant les résultats d'alignement avec le filtre *ExpLoc-P* et les paramètres d'optimisation fournis par les auteurs.
- *ExpLocPNoOpt* : en gris, la courbe présentant les résultats d'alignement avec le filtre *ExpLoc-P* sans paramètre.

On observe sur la Figure 3.64 que toutes les méthodes, à l'exception d'*ExpLocPNoOpt*, fonctionnent bien et présentent des comportements relativement similaires. Cependant pour les alignements de référence présentant une similarité en séquence comprise entre 0,6 et 0,8 (soit entre 60% et 80% de similarité en séquence), lorsque l'on utilise *RNA-unchained* avec les critères de sélections sur les graines, soit qu'elles soient composées d'au moins deux types d'éléments structuraux différents, on obtient de meilleurs résultats d'alignement, et même de meilleurs alignements que *LocARNA* seul. Cela montre que l'ajout d'ancres de haute qualité, même si elles ne recouvrent qu'une faible partie des séquences ARN considérées (voir la Figure 3.63), peut améliorer significativement la qualité des alignements. Cependant la définition de hits trop stringent, comme c'est le cas avec l'option $91MFE\_r$ par exemple, génère un taux de couverture par les ancres très faible. Ce qui a pour conséquence que la plupart des alignements ne présentent pas d'ancre et donc pas de contrainte à fournir à *LocARNA*. Ces alignements reposeront alors sur l'alignement seul par *LocARNA* des séquences.

3.4.3 Impact des Options des Graines

(i) Influence de la Couverture des Ancres

Afin de comprendre les différences de qualité entre les alignements obtenus, le pourcentage de contraintes communes entre les deux méthodes, *RNA-unchained* et *ExpLoc-P*, a été calculé. Les méthodes $91MFE$, $91MFE\_rfb$ et $91MFE\_rfbepc$ présentent entre 50 et 80% de contraintes communes avec *ExpLoc-P* excepté dans l'intervalle de $60\% - 80\%$ de similarité où le pourcentage de contraintes communes



(a) Analyse de la couverture des graines

Figure *3.65 – Pourcentage de contraintes communes* à ExpLoc-P.

est inférieur à 50%. De même, la méthode *91MFE_r* présente moins de 30% de contraintes communes avec *ExpLocP* en decà de 90% de similarité en séquence des alignements.

Il est intéressant de noter que notre modèle de graines (combiné aux deux options) permet une couverture des séquences ARN comparable à la couverture des EPM (voir Figure 3.65). Lorsque l'on met en parallèle les Figures 3.65 et 3.63, on observe que la qualité des alignements de *91MFE_r* est la plus proche de celle de *LocARNA*, cependant c'est aussi celle qui présente le plus faible taux de contraintes communes avec *ExpLoc-P*. Ceci implique que le peu de hits de $91MFE\_r$ semble plus significatif que ceux d'*ExpLoc-P*.

(ii) Influence du nombre d'alignements couverts

L'ensemble des alignements de BraliBase2.1 n'est pas couvert par des hits. Ainsi, si un alignement ne présente pas de hits entre les deux séquences qui le compose, *LocARNA* est appliqué seul et sans contrainte sur les séquences de cet alignement. *LocARNA* n'ayant alors aucune contrainte sur laquelle baser son alignement, ses temps de calculs ne sont pas améliorés. Il est donc important de comparer le nombre d'alignements couverts par des hits selon la méthode et les options choisies.

La Figure 3.66 présente pour les deux méthodes étudiées, *ExpLocP* et *RNA-unchained*, le nombre d'alignements couverts par des hits en fonction du taux de couverture. On note que la première barre de l'histogramme (à 0) correspond au nombre d'alignements non couverts par les hits. Cette valeur est donnée explicitement à la suite du nombre d'alignements couverts par au moins un hit dans le titre de chaque histogramme.

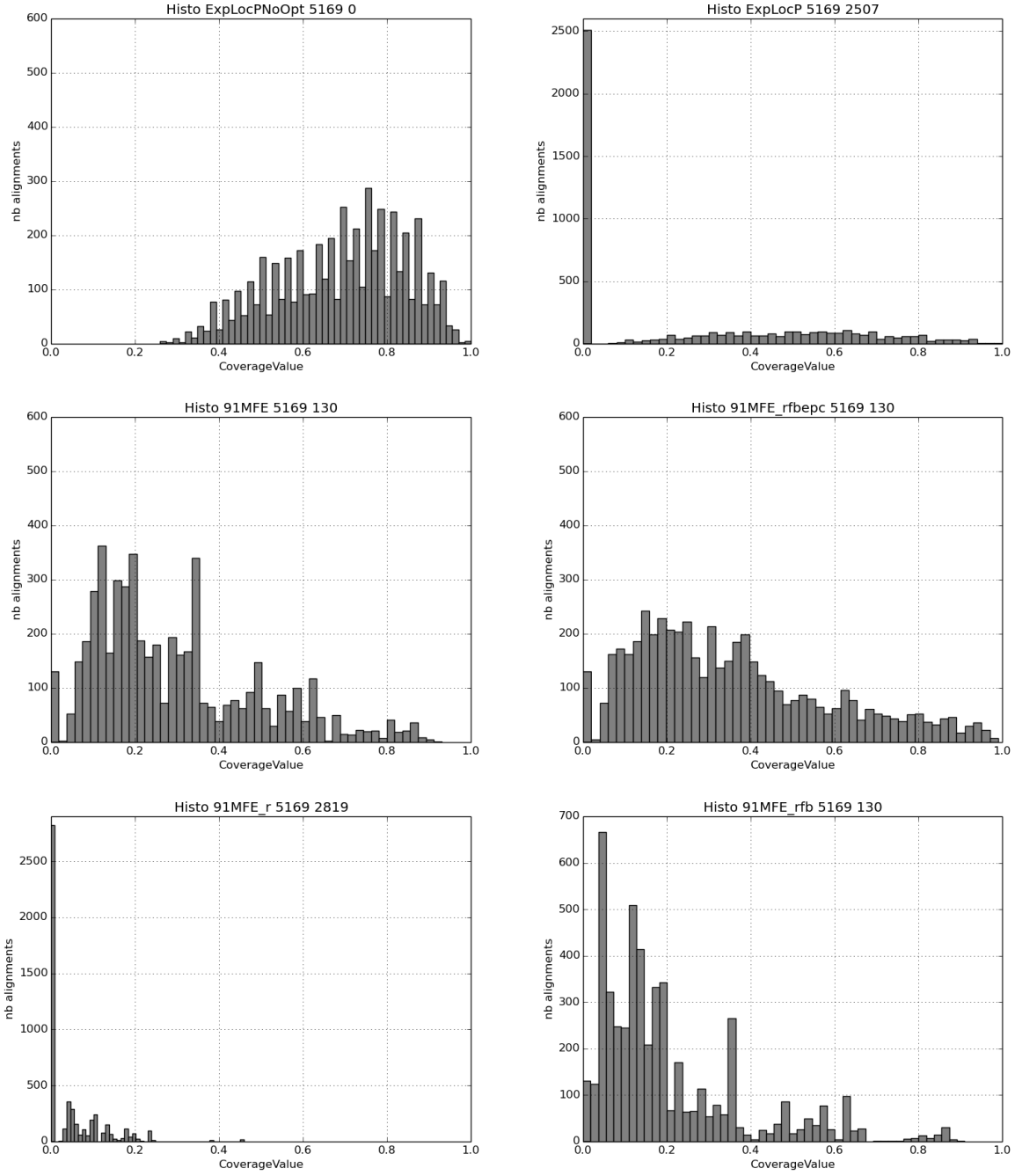


FIGURE *3.66 – Nombre d'alignements couverts par des hits chaînés selon la méthode employée.*

Pour *RNA-unchained*, on observe que le nombre d'alignements sans hit pour les options $91MFE$, $91MFE\_rfb$ et $91MFE\_rfbepc$ est le même puisque dans ces trois cas aucun des hits n'est écarté avant le chainage si cela entraine l'absence de

hits dans l'alignement. La différence entre ces trois options réside dans la répartition des alignements couverts. En effet, les options combinées rfb et *epc* permettent d'augmenter le nombre d'alignements avec une meilleure couverture. Au contraire, l'option $91MFE\_r$ écarte certains hits même si cela signifie qu'il n'y aura plus de hits entre les séquences. On observe alors un grand nombre (2 819) d'alignements non couverts et la majorité des alignements présente une faible couverture par les hits. Cependant si on rapproche ces résultats des précédents résultats présentés (Figure 3.63) cela laisse entendre que l'option r permet un chainage plus stringent grâce à des hits plus pertinents. Mais ces résultats peuvent être modérés par le grand nombre d'alignements non couverts. Le nombre d'alignements couverts avec les options rfb et *epc* permettent de combiner la pertinence des graines de l'option r avec le taux de couverture de base des hits $91MFE$. De plus, le nombre d'alignements présentant un fort taux de couverture est amélioré par l'option *epc* qui permet d'étendre les hits pré-existants.

De manière similaire à l'option r, *ExpLocP* ne couvre pas non plus la totalité des alignements (2 507 ne sont pas couverts). On peut donc en déduire les mêmes résultats à savoir que les EPM d'*ExpLocP* sont suffisament pertinentes pour améliorer la qualité des alignements, cependant de nombreux alignements ne sont pas couverts. Au contraire *ExpLocPNoOpt* génère des EPM qui couvrent la totalité des alignements mais la qualité de ces alignements est faible (Figure 3.64). En effet, de nombreuses EPM identifiées ne sont pas suffisament significatives pour générer un signal de chainage fort ce qui se traduit par des alignements erronés.

(iii) Influence du Temps de Calcul

Différents temps de calcul ont été répertoriés pour différentes méthodes présentées dans les sections précédentes. On remarque à cette occasion que les temps de calcul de *LocARNA* et *ExpLocP* ne sont pas aussi importants que ceux présentés dans (Schmiedl et al., 2012). Cependant les temps de calcul que nous obtenons pour *ExpLocP* sont comparables à (Schmiedl et al., 2012). La différence observée provient des récentes améliorations apportées à *LocARNA*.

Temps de Calcul	Hits/chainage	Alignement des gaps	Total
LocARNA	0	9,022	9,022
ExpLocP	1,492	6,070	7,562
$91MFE$	3,386	4,563	7,949
$91MFE\_r$	3,157	6,242	9,399
$91MFE\_rfb$	3,329	5,955	9,284
$91MFE\_rfbepc$	3,283	4,510	7,793

Table 3.3 – *Temps de calcul (en secondes). Le temps de calcul requis pour la construction de l'index n'est pas inclue mais prend moins d'une minute. Les expériences ont été réalisées sur un serveur à double processeurs Intel Xeon 3.3GHz. Les étapes d'indexation des graines, de recherche des hits et de chaînage sont implémentées en Java.*

On note que des valeurs élevées en structure (paramètre l), et plus particulièrement combinées à l'option requérant deux types d'éléments structuraux différents, comme par exemple pour $91MFE\_r$, mènent à des alignements de bonne qualité mais à de faibles taux de couverture par les graines et à des temps de calculs pour la phase d'alignement des segments entre les graines très élevés.

L'analyse de $91MFE$ montre un taux de couverture par les ancres relativement élevé, ce qui implique que *LocARNA* tire énormément avantage de ce taux de couverture dans le calcul de l'alignement final puisque le temps de calcul est divisé par deux. Cependant, la justesse des alignements est significativement inférieure à celle obtenue par les autres méthodes analysées. Ceci peut être expliqué par la présence dans l'ancre finale de hits présentant de faibles informations du point de vue structural (comme par exemple un segment conservé de bases non appariées). Ces segments sont alors des segments *faux positifs* sélectionnés au cours du chaînage et faussant *LocARNA* au cours de la phase finale d'alignement exact basée sur l'ancre fournie.

Entre ces deux méthodes extrêmes, on peut observer que l'optimisation des graines que nous avons implémentée et expérimentée avec $91MFE\_rfbepc$ (voir le paragraphe (iii)) offre un bon compromis en présentant des résultats d'alignement de qualité et de meilleurs temps de calculs, en particulier concernant la phase d'alignement.

Enfin, la seconde optimisation proposée dans *RNA-unchained*, l'extension des ancres basée sur le LCS (voir le paragraphe (ii)), améliore significativement le taux de couverture par l'ancre sans impacter significativement la qualité des alignements obtenus. Cela se reflète dans les valeurs de SPS obtenue par $91MFE\_rfbepc$ qui sont particulièrement proches des valeurs obtenues par $91MFE\_rfb$ alors que le taux de couverture est le plus élevé parmi toutes nos versions. En conclusion, le gain en temps de calcul de l'alignement en comparaison avec *LocARNA* est maximal (divisé par 2).

3.5 Conclusion & Perspectives

RNA-unchained permet la comparaison d'un ARN avec un ensemble d'ARN, en se basant sur les notions de graines, d'indexation de graines et de chaînage de hits. Les points clefs de notre approche sont donc : un modèle de graines basé à la fois sur la séquence et sur la structure et un algorithme de chaînage de graines rapide (sub-cubique). La capacité d'indexer rapidement et de retrouver efficacement les hits entre une séquence requête et un jeu de données cibles est un point important de notre méthode puisque l'indexation de l'ensemble des graines des séquences cibles est réalisée une unique fois en un temps linéaire par rapport à la somme des tailles de ces séquences. En effet, pour cette étape d'indexation seules les valeurs des paramètres l et d des graines importent puisqu'elles déterminent le nombre de clefs de la table de hachage. Les expériences réalisées avec Bralibase2.1 montrent clairement que *RNA-unchained* obtient des résultats de meilleure qualité que les méthodes actuelles, avec

des temps de calculs comparables.

Le modèle de graines introduit diffère significativement du modèle des *EPM* d'*ExpaRNA*. Il est intéressant de remarquer que notre modèle de graines associé aux deux améliorations proposées génère une couverture des séquences comparable à celle des EPM. Cela montre que ces deux modèles semblent capables d'identifier des motifs structuraux conservés importants. Cependant, pour des similarités en séquences comprises entre 60 et 80%, on observe une différence significative (*RNA-unchained* améliore la qualité des alignements par rapport à *ExpLoc-P* et *LocARNA*). De manière plus générale, nos résultats ainsi que ceux centrés autour de la notion d'EPM, suggèrent que les approches de chaînage de hits méritent d'être explorées aussi bien pour leurs modèles de graines que pour leurs algorithmes de chaînage. En particulier, contrairement aux graines sur les séquences qui ont été largement étudiées (Brown, 2008), des analyses statistiques portant sur les modèles de graines sur les ARN manquent.

La différence majeure entre notre approche et celles d'*ExpLoc-P* et *LocARNA* repose sur la méthode employée pour prendre en compte la structure secondaire des ARN. *ExpLoc-P* et *LocARNA* suivent une approche globale basée sur les probabilité d'appariement des bases selon la distribution de Boltzmann, alors que notre approche se base sur les structures secondaire MFE. Nous avons étudié des approches intermédiaires basées sur le repliement en structure secondaire avec *RNAsubopt* de Lorenz et al. (2011) et *RNAshapes* de Steffen et al. (2006). Cependant nous avons observé que la structure MFE permet de produire des résultats de meilleure qualité tout en minimisant les temps de calcul. Cette concordance apparente entre deux approches différentes suggère à nouveau que les graines intégrant une dimension structurale, c'est-à-dire soutenant l'idée de la conservation de motifs en séquences et structures, méritent d'être étudiées.

Un aspect important de *RNA-unchained* concerne l'utilisation de *LocARNA* pour l'alignement contraint des gaps. Comme les gaps correspondent aux segments dans lesquels aucun motif structural n'est conservé, cela permet de réduire l'impact du choix de la structure MFE. En effet, celle-ci est alors uniquement utilisée pour détecter les hits et calculer l'ancre, ce qui peut constituer une des raisons pouvant expliquer la concordance entre les deux approches. Cependant, cette partie du pipeline est la plus coûteuse du point de vue du temps de calcul, ce qui est également le cas pour *ExpLoc-P*, puisque *LocARNA* est basé sur le modèle de repliement et d'alignement simultané de Sankoff. Les résultats obtenus avec l'option d'extension des ancres suggèrent qu'une approche hiérarchique, qui prend en considération des motifs en séquence ou structure moins conservés au sein des gaps, permet de réduire la taille des segments pour lesquels on calcule un alignement exact et serait une approche efficace. Il est même possible de se demander si pour certaines applications pour lesquelles un alignement exact n'est pas nécessaire (telles que par exemple les premières étapes de regroupement initial au cours de l'analyse du structurome ARN d'un génome complet), l'approche décrite ci-avant ne serait pas suffisante.

Concernant *RNA-unchained*, sa dernière version est à ce jour disponible sous la

forme d'une suite de programmes écrits en java. Une nouvelle version en $C++$ est en cours d'implémentation. On peut donc supposer que les temps de calculs seront améliorés par cette version.

Le modèle de graines utilisé, les graines (l, d) centrée, est un sous-ensemble du modèle des « graines à trous » (ou « spaced seeds »). Dans ce modèle, les mésappariements en séquence (au nombre de $l - 2d$) ne sont pas uniquement localisés sur les extrémités de la graine, il sont répartis sur chacune des positions possibles sur l. On obtient alors pour un couple (l, d) de paramètres $n' = \binom{l}{l-2d}$ graines différentes. Il serait intéressant de voir l'impact de l'intégration de cette nouvelle variable (la position des gaps) sur la qualité des alignements produits.

Tout comme dans le chaînage en séquence, entre deux hits consécutifs chaînables on peut observer une région non contrainte, ou gap. Plus un gap est long moins les deux séquences comparées sont similaires. Au cours du chaînage dans les arborescences ces gaps ne sont pas pris en compte lors du calcul de l'ancre. Il serait donc intéressant de se pencher sur le problème de l'introduction de la notion de gap dans les arborescences comme cela a été fait dans les séquences par Ohlebusch and Abouelhoda (2006).

La structure d'indexation actuellement utilisée pose certains problèmes mémoires puisque la plupart des tables constituant l'index sont creuses. Il serait donc pertinent de rechercher une structure d'indexation plus efficace et plus compacte. De plus, il est réaliste de penser qu'un certain nombre de graines identifiées soit des motifs récurrents sur la majorité des séquences de la base de données, d'autant plus si le modèle de graines utilisé est de petite taille. En effet, présente sur une grande partie des séquences de l'index, ce type de graines n'apporte aucune information supplémentaire voir même, au contraire, peut induire en erreur. Une analyse statistique serait utile afin d'identifier les graines récurrentes sur-représentées dans la base d'indexation et ainsi pouvoir les éliminer de toute future analyse.

Bibliographie

Allali, J., Chauve, C., Ferraro, P., and Gaillard, A. (2012). Efficient chaining of seeds in ordered trees. *J. of Discrete Algorithms*, 14 :107–118.

Allali, J., d'Aubenton Carafa, Y., Chauve, C., Denise, A., Drevet, C., Ferraro, P., Gautheret, D., Herrbach, C., Leclerc, F., De Monte, A., et al. (2008). Benchmarking rna secondary structure comparison algorithms. *Actes des Journées Ouvertes de Biologie, Informatique et Mathématiques-JOBIM'08*, pages 67–68.

Allali, J. and Sagot, M. (2008). A multiple layer model to compare rna secondary structures. *Software : Practice and Experience*, 38(8) :775–792.

Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1) :217–239.

Blin, G. and Touzet, H. (2006). How to compare arc-annotated sequences : The alignment hierarchy. In *String Processing and Information Retrieval*, pages 291–303. Springer.

Brown, D. (2008). *Bioinformatics algorithms : techniques and applications*, volume 3. Wiley-interscience.

Demaine, E. D., Mozes, S., Rossman, B., and Weimann, O. (2009). An optimal decomposition algorithm for tree edit distance. *ACM Transactions on Algorithms (TALG)*, 6(1) :2.

Do, C., Mahabhashyam, M., Brudno, M., and Batzoglou, S. (2005). Probcons : Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2) :330–340.

Evans, P. (1999). Finding common subsequences with arcs and pseudoknots. In *Combinatorial Pattern Matching*, pages 270–280. Springer.

Havgaard, J., Torarinsson, E., and Gorodkin, J. (2007). Fast pairwise structural rna alignments by pruning of the dynamical programming matrix. *PLOS computational biology*, 3(10) :e193.

Heyne, S., Will, S., Beckstette, M., and Backofen, R. (2009). Lightweight comparison of rnas based on exact sequence-structure matches. *Bioinformatics*, page btp065.

Hochsmann, M., Toller, T., Giegerich, R., and Kurtz, S. (2003). Local similarity in rna secondary structures. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 159–168. IEEE.

Hofacker, I. and Stadler, P. (2010). Rnaz 2.0 : improved noncoding rna detection. In *Pacific Symposium on Biocomputing*, volume 15, pages 69–79. World Scientific.

Hogeweg, P. and Hesper, B. (1984). Energy directed folding of rna sequences. *Nucleic acids research*, 12(1Part1) :67–74.

Jiang, T., Wang, L., and Zhang, K. (1995). Alignment of trees-an alternative to tree edit. *Theoretical Computer Science*, 143(1) :137–148.

Lorenz, R., Bernhart, S., Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P., Hofacker, I., et al. (2011). Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1) :26.

Lyngsø, R. B. and Pedersen, C. N. (2000). Rna pseudoknot prediction in energy-based models. *Journal of computational biology*, 7(3–4) :409–427.

Nagel, J., Gultyaev, A., Gerdes, K., and Pleij, C. (1999). Metastable structures and refolding kinetics in hok mrna of plasmid r1. *RNA*, 5(11) :1408–1418.

Ohlebusch, E. and Abouelhoda, M. I. (2006). Chaining algorithms and applications in comparative genomics. *Handbook of Computational Molecular Biology*.

Parisien, M. and Major, F. (2008). The mc-fold and mc-sym pipeline infers rna structure from sequence data. *Nature*, 452(7183) :51–55.

Peattie, D. A. and Gilbert, W. (1980). Chemical probes for higher-order structure in rna. *Proceedings of the National Academy of Sciences*, 77(8) :4679–4682.

Perriquet, O., Touzet, H., and Dauchet, M. (2003). Finding the common structure shared by two homologous rnas. *Bioinformatics*, 19(1) :108–116.

Sankoff, D. (1985). Simultaneous solution of the rna folding, alignment and proto-sequence problems. *SIAM Journal on Applied Mathematics*, 45(5) :810–825.

Schirmer, S. and Giegerich, R. (2013). Forest alignment with affine gaps and anchors, applied in rna structure comparison. *Theoretical Computer Science*, 483 :51–67.

Schmiedl, C., Möhl, M., Heyne, S., Amit, M., Landau, G., Will, S., and Backofen, R. (2012). Exact pattern matching for rna structure ensembles. In *Research in Computational Molecular Biology*, pages 245–260. Springer.

Shapiro, B. and Zhang, K. (1990). Comparing multiple rna secondary structures using tree comparisons. *Computer applications in the biosciences : CABIOS*, 6(4) :309–318.

Steffen, P., Voß, B., Rehmsmeier, M., Reeder, J., and Giegerich, R. (2006). Rnashapes : an integrated rna analysis package based on abstract shapes. *Bioinformatics*, 22(4) :500–503.

Touzet, H. and Perriquet, O. (2004). Carnac : folding families of related rnas. *Nucleic acids research*, 32(suppl 2) :W142–W145.

Will, S., Reiche, K., Hofacker, I., Stadler, P., and Backofen, R. (2007). Inferring noncoding rna families and classes by means of genome-scale structure-based clustering. *PLoS computational biology*, 3(4) :e65.

Wilm, A., Mainz, I., and Steger, G. (2006). An enhanced rna alignment benchmark for sequence alignment programs. *Algorithms for molecular biology*, 1(1) :19.

Zuker, M. and Sankoff, D. (1984). Rna secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4) :591–621.

Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1) :133–148.

Chapitre 4

Méthode d'Identification et de Caractérisation des Pseudogènes au sein de Génomes Procaryotes

La vinification est caractérisée par deux fermentations, qui doivent être maîtrisées par le vinificateur, impliquant la flore microbienne naturellement présente dans le moût de raisin. Tout d'abord les levures assurent la fermentation alcoolique qui voit la conversion des sucres en alcool. Ensuite, les bactéries lactiques réalisent la fermentation malolactique, qui consiste en la dégradation de l'acide malique en acide lactique. En plus de diminuer l'acidité, la fermentation malolactique s'accompagne, entre autres, d'une stabilisation microbiologique du vin. La fermentation malolactique est un processus bénéfique et nécessaire à la production de la plupart des vins. Toutefois, la capacité des bactéries lactiques autochtones à survivre et à se développer dans ces conditions physico-chimiques particulièrement hostiles (forte concentration en éthanol ($> 10\%$), pH bas (3-4), appauvrissement en éléments nutritionnels) détermine son bon déroulement. Dans ces conditions, les bactéries lactiques les plus résistantes sont naturellement sélectionnées ; le plus souvent *Oenococcus oeni* devient l'espèce majoritaire, et du même coup le déclencheur et principal acteur de la fermentation malolactique.

Oenococcus oeni apparaît comme « la bactérie la mieux adaptée aux conditions de vinification ». Cependant, une grande diversité de phénotypes est naturellement observée aux niveaux des souches. Celles-ci diffèrent notamment par leur tolérance à certains types de vin, leur efficacité fermentaire ou encore la production d'arômes via différentes voies métaboliques. En conséquence du caractère variable et imprévisible de la flore lactique, le recours à l'inoculation de souches d'*Oenococcus oeni* sélectionnées est devenu une pratique courante pour contrôler ou amorcer une fermentation malolactique tardive. L'organisation des populations du genre *Oenococcus* est connue et panmictique (Bilhère et al., 2009). Cependant, ces souches présentent de fortes variations génotypiques (Borneman et al., 2010; Bon et al., 2009) et des

variations phénotypiques hétérogènes (Bilhère et al., 2009; Bartowsky and Borneman, 2011), ce qui rend ces espèces bactériennes hautement spécialisées difficiles à « domestiquer » par les viticulteurs.

À ce jour, les génomes des différentes souches identifiées (plus de 200) de l'espèce sont en cours de séquençage ou d'annotation. En particulier, les génomes de deux souches aux performances oenologiques antagonistes ont été entièrement séquencés et circularisés : la souche PSU-1 et la souche BAA-1163. Alors que PSU-1 compose un levain industriel, BAA-1163 ne présente aucune propriété bénéfique pouvant justifier son incorporation à la composition d'un levain. En parallèle, les logiciels d'inférence de présence de gènes permettent l'automatisation de la cartographie génique. Cependant de nouvelles entités génomiques récemment identifiées, les pseudogènes, pouvant également avoir un rôle fonctionnel dans l'expression génique, faussent ces prédictions automatiques. Il apparaît donc comme nécessaire de s'attacher à la détection de cette nouvelle kyrielle fonctionnelle pour identifier son hypothétique implication dans les performances de la souche.

Les variations du contenu des gènes, la plasticité des génomes, sont connues pour être un point clef de l'évolution des génomes. L'étude de ces variations est une étape importante pour la compréhension de l'évolution et de l'adaptation des génomes à une niche écologique (Makarova and Koonin, 2007). Les modifications de mode de vie des bactéries ou les conditions de vie difficiles peuvent mener à une variabilité du génome, à la perte ou l'acquisition de fonctions.

Les pseudogènes sont des séquences qui dérivent de séquences géniques ayant codé pour des protéines et perdu cette capacité suite à des altérations. En proportions moins importantes que dans les génomes eucaryotes, ils sont également présents dans les génomes procaryotes. Les pseudogènes proviennent de divers mécanismes génétiques et évolutionnaires qui altèrent la transcription ou la traduction de ces séquences (Rouchka and Cha, 2009). La proportion des pseudogènes peut énormément varier d'un organisme à un autre (entre 1% et 5% mais rarement au delà de 40%), selon le mode de vie (individuel ou en association) et les propriétés du génome (taux de duplication, mutation, deletion et de transfert latéral) (Liu et al., 2004).

Outre l'avantage phylogénétique soulevé, un intérêt algorithmique est également à souligner. En effet, l'automatisation du décèlement des pseudogènes permettrait de pallier la confusion qu'ils engendrent lors de la détection automatique des gènes, et par là même de l'améliorer. Les séquences pseudogéniques constituent donc des leurres à deux niveaux : pour les logiciels d'annotation en raison de leur séquence proche de celles des gènes codant pour des protéines et via leur ARN qui ainsi permet la régulation de l'expression génique (Deroin, 2010).

Nous étudierons le pseudome des génomes bactériens du vin afin de reconstruire et de modéliser l'histoire évolutive des séquences géniques en nous focalisant sur les génomes du genre *Oenococcus* qui, de plus, présentent un intérêt biotechnologique. D'autre part, ces souches présentent un génome hautement compact et des séquences pseudogéniques. L'analyse de ces dernières présente un double intérêt tant du point de vue fonctionnel, au sein de la cellule, que du point de vue *in silico*, au niveau de

l'annotation des objets géniques.

4.1 Nomenclature Systématique des Objets Génétiques

4.1.1 Nécessité d'une Nomenclature Commune

Absence d'une nomenclature harmonisée Les différents génomes d'*O. oeni* ont été séquencés par différents laboratoires. Ainsi chacun des génomes annotés présente une nomenclature différente adoptée par le laboratoire dont il est issu.

Il n'existe actuellement aucune nomenclature commune aux génomes d'*O. oeni* ou même aux génomes bactériens. En outre, il est difficile de trouver des informations sur les conventions de nomenclature systématique des entités génomiques. De plus, la plupart des sites internet n'explicitent pas leurs conventions. Seuls certains incluent un paragraphe abordant le système de nomenclature sans pour autant le détailler.

Les conventions établies ont ici été essentiellement déduites du système de nomenclature des gènes des données des séquences publiques (EMBL) ainsi que du système de nomenclature des levures mis en place par (Durrens and Sherman, 2005).

Nécessité d'une nomenclature (bactérienne) normalisée, explicite et extensible L'augmentation du nombre de génomes bactériens séquencés et la diversité de nommage des entités génomiques identifiées qui en découle sont à l'origine de l'émergence du besoin d'établissement d'une nomenclature unique. En effet, actuellement, des noms arbitraires sans règle nécessairement établie leur sont attribués : aucune identification instantanée de l'espèce n'est alors possible. Il apparaît comme nécessaire d'adopter un système simple, stable, sans ambiguïté et extensible de nomenclature de ces éléments génomiques. Pour cela des règles précises doivent être érigées.

Afin d'analyser les séquences géniques, pseudogéniques et intergéniques des bactéries lactiques et de les comparer entre elles, au sein d'un même génome ou dans deux génomes différents, une sémantique de nomenclature unique est nécessaire.

L'assemblage de nombreux génomes bactériens étant en cours d'étude ou en phase d'achèvement, fini ou presque, les entités actuellement identifiées (CDS, ARN, pseudogènes) constituent des ancres stables pour le génome étudié et limitent le nombre d'éléments nouveaux pouvant s'insérer.

4.1.2 Élaboration d'une Nomenclature Systématique

Cette nomenclature doit prendre appui sur des caractéristiques stables et uniques de la séquence nucléotidique. Effectivement, si certaines entités telles que les CDS sont facilement détectées d'autres comme les promoteurs le sont bien moins. En outre, les caractéristiques sémantiques (comme la fonction par exemple) ne sont

pas appropriées pour fonder un système stable de nomenclature. En effet, ces interprétations peuvent évoluer avec le temps. En outre, certains éléments génomiques comme les gènes codant ou les ARN peuvent présenter plusieurs copies, ce qui peut engendrer des confusions. Il est donc nécessaire d'établir un système de nomenclature utile, et pouvant être utilisé lors d'analyses *in silico*, des éléments composants l'ADN des bactéries lactiques.

Nomenclature élaborée L'ensemble des informations rapportées a permis d'établir une nomenclature systématique respectant les caractéristiques qui suivent. La nomenclature doit :

- spécifier l'espèce ainsi que le contig [1] auquel appartient l'entité. Un nombre unique permet d'identifier précisément l'entité.
- expliciter le type de l'élément en question (CDS, ARN, ...).
- intégrer l'ensemble des entités génomiques présentes qu'elles soient transcrites ou non ou traduites ou non.
- être extensible pour autoriser l'insertion de nouvelles entités mises en évidence tout en conservant l'ordre relatif des éléments.
- s'affranchir des caractères subjectifs comme par exemple la fonction ou les relations d'homologies.

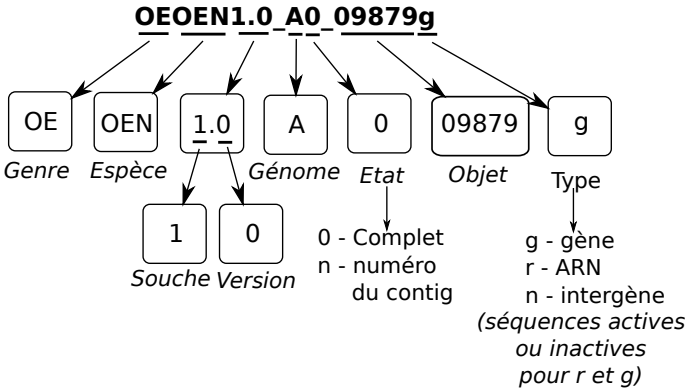


FIGURE 4.67 – *Nomenclature des séquences proposée*

La syntaxe qui suit a alors été développée en respectant l'ensemble de ces critères (voir Figure 4.67) :

- genre := 2 lettres
- espèce := 3 lettres
- souche := 1 chiffre
- version := 1 chiffre
- genome := 1 majuscule ('A'|'B'|'C')

1. Ensemble de fragments d'ADN clonés chevauchants pouvant être séquencés et assemblés pour représenter une région définie du chromosome ou du génome duquel ils ont été obtenus. La définition des contigs est une étape nécessaire pour l'assemblage de séquences entières d'un génome

- contig := 1 chiffre
- objet := 5 chiffres
- type := 1 minuscule ('g'|'r'|'n')

Cette syntaxe prend en considération les informations taxonomiques, génomiques et géniques.

genre et **espèce** font référence au genre et à l'espèce dont est issue la séquence. Le code à 5 lettres reprend les deux premières lettres du genre de l'organisme et les trois premières de l'espèce en question. Cette combinaison permet une précision suffisante pour éviter toute synonymie au sein des bactéries lactiques tout en limitant le nombre de caractères totaux.

souche donne la souche de la bactérie lactique. Pour la retrouver un tableau de correspondances classé par espèce puis par souche, dans l'ordre alphabétique est disponible.

Tout comme pour la **souche**, le chiffre codant pour la **version** est présent dans la table de correspondances.

genome fait référence au support de l'information génique. Chez les procaryotes, le support principal de l'information est le chromosome circulaire bactérien symbolisé par la lettre A. Il est également possible de rencontrer des génomes extrachromosomiques tels que : des plasmides (B) ou des phages (C).

Les génomes séquencés ne sont pas toujours complets et certains demeurent à l'état de draft, c'est-à-dire sous la forme d'un puzzle de plusieurs contigs non assemblés. Le chiffre du **contig** fait donc référence au numéro du contig (0 signifiant que l'assemblage du génome est complet).

objet est un nombre à cinq chiffres qui fait référence à l'objet génique proprement dit. Ce nombre va croissant de la gauche vers la droite de la séquence, ce qui permet d'organiser dans l'espace ces éléments.

- La numérotation s'incrémente de 11 entre deux éléments géniques, ce qui permet l'insertion de tout nouvel élément, en cas de reséquençage ou de réannotation du génome, sans bouleverser la chronologie de cette numérotation dans l'espace. L'intérêt de l'incrément de 11 est que tous ses chiffres sont significatifs (avec un incrément de 10, le chiffre des unités serait égal à 0 la plupart du temps).
- Tout élément inséré se voit attribuer un nombre interpolé compris entre celui des deux éléments qui l'encadrent.
- La numérotation commence a 00001 puisque c'est une CDS identifiée qui est prise pour « origine » du génome.

type renseigne le type de séquence de l'objet grâce à un système de codage à une lettre. Pour éviter toute confusion les lettres i et o (pouvant être confondues respectivement avec le 1 et le 0) ne sont pas utilisées.

- g : fait référence à toutes les séquences codant (CDS) ou ayant codé (pseudogènes) pour une protéine.
- r : fait référence aux séquences codant pour des ARN.
- n : fait références aux morceaux de séquences séparant deux objets géniques.

Cas des génomes incomplets L'assemblage de certains génomes demeure parfois incomplet. Malgré leur état de draft, ces génomes peuvent quand même répondre aux règles établies, en particulier grâce au champ **contig**. De plus, si ultérieurement deux contigs sont assemblés, le numéro du contig le plus petit est conservé et les éléments du second se voient attribuer un nombre dans la continuité de la numérotation du premier. Mais surtout, ce nouvel assemblage constitue une nouvelle version du génome de la souche et porte donc un nouvel identifiant de **version**.

4.2 Caractérisation des Pseudogènes

4.2.1 Caractérisation

Les répertoires géniques d'un génome proviennent d'un équilibre subtil entre dérive génétique, perte et gain de matériel génétique. Il est donc important pour comprendre l'évolution de la plasticité génique de recenser les gènes codants des protéines (CDS) et ceux ayant ancestralement codé des protéines (pseudogènes) et de les cartographier.

La pseudogénisation d'objets géniques peut provenir d'événements divers allant d'une mutation ponctuelle isolée ou d'une accumulation de mutations affectant la CDS, les promoteur ou terminateur, à des événements de duplication pouvant conduire à la genèse de copies surnuméraires plus ou moins parfaites de l'objet dupliqué mais aussi à des pseudogénisations collatérales par fusion ou fission.

L'évaluation dans les génomes du degré de redondance des objets géniques par le biais du nombre de leurs copies permet de dresser un inventaire des paralogues et par là même de définir les familles de gènes, intacts ou interrompus. On distingue ainsi les pseudogènes unitaires, qui ne présentent aucun paralogue, des pseudogènes appartenant à des familles multigéniques.

La localisation même de ces objets au sein du génome par le biais d'une cartographie peut permettre la mise en évidence d'une distribution chromosomique non aléatoire, diffuse ou organisée en territoires de pseudogénisation plus ou moins denses.

Les différentes classes de pseudogènes seront également caractérisées par l'analyse comparée de leur biais compositionnel.

(i) Classes Pseudogéniques

L'état pseudogénique de séquences implique une absence d'expression ou de codage pour des protéines et résulte de déficiences acquises (voir Figure 4.68) dues à des insertions aberrantes de codons STOP, de cassures physiques...(*cf.* Section 1.3.2.(ii)).

Concernant les bactéries lactiques, la réduction et la spécialisation des génomes sont également à l'origine de certains « traits génétiques » qui sont le reflet direct de

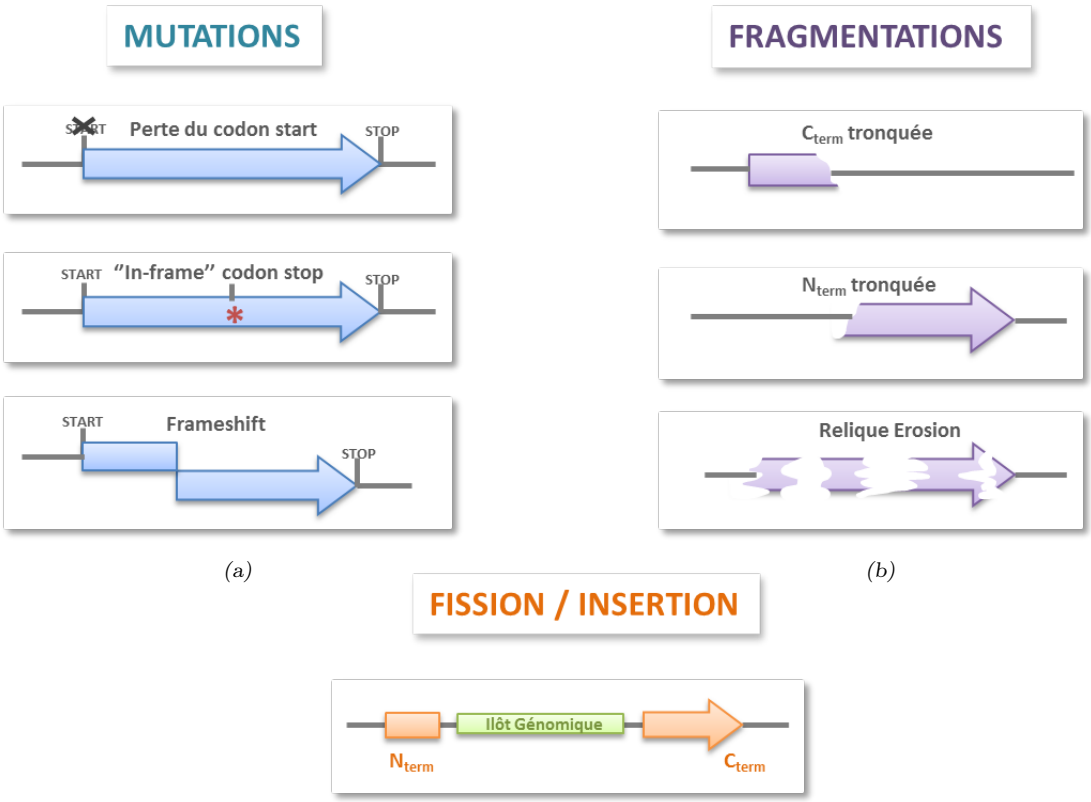


Figure 4.68 – *Différents types de pseudogènes pris en compte au cours de l'analyse.*

l'adaptation de ces microorganismes à leur environnement. Le séquençage et l'analyse des génomes complets a permis notamment de mettre en évidence un nombre important de pseudogènes chez plusieurs bactéries lactiques (voir la Table 1.2). Cette observation concerne principalement les espèces colonisant des niches écologiques spécialisées comme *Lactobacillus bulgaricus* (270 et 183 pseudogènes respectivement pour les souches ATCC 11842 et ATCC BAA-365 (Van de Guchte et al., 2006; Makarova et al., 2006)) ou *O. oeni* (122 pseudogènes (Makarova et al., 2006)).

(ii) Modélisation des États Évolutifs

Modélisation de la pseudogénisation : « dévolution génique » Une fois les pseudogènes d'un génome identifiés puis caractérisés, l'ensemble des données rassemblées informe sur les processus de pseudogénisation. Il peut alors être envisageable d'établir des relations entre les diverses informations recueillies pour formaliser les voies de pseudogénisation. Un modèle des voies de pseudogénisation des gènes peut être établi.

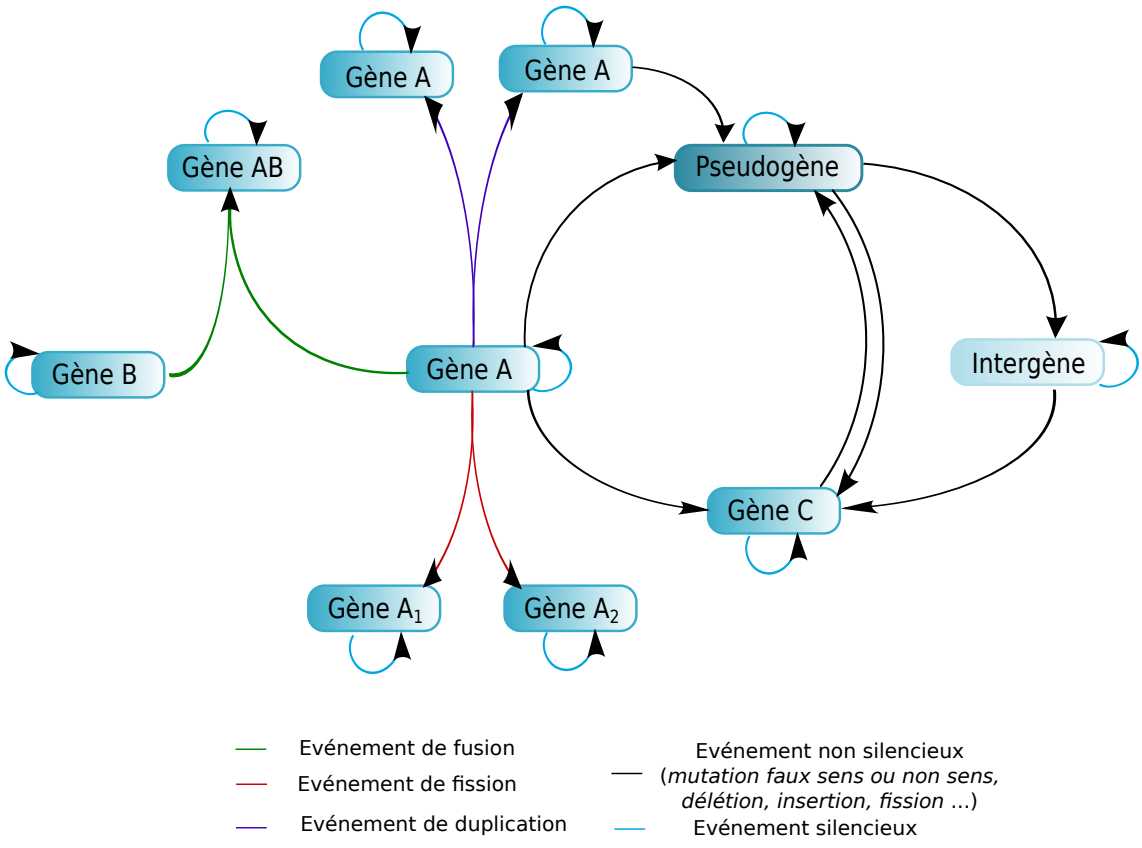


FIGURE 4.69 – *Modèle inférant l'évolution des objets géniques au sein d'un génome G_A et permettant l'analyse comparative des objets pseudogéniques de ce génome avec un second génome G_B.*

À partir de l'étude de génomes bactériens d'*Oenococcus* des similarités et divergences concernant les pseudogènes peuvent être discernées. Cela peut permettre l'élaboration d'un modèle de la pseudogénisation des séquences bactériennes. L'étude des données concernant l'évolution des gènes ainsi que les connaissances sur les pseudogènes permettent d'établir un premier schéma de l'évolution possible des gènes codant ou ayant codé des protéines (voir Figure 4.69).

4.2.2 Recensement

Dans le but de pouvoir analyser les séquences pseudogéniques et de comprendre les mécanismes qui les régissent il faut avant tout caractériser ces séquences. En effet, certains pseudogènes pourraient ne pas être complets ou avoir été mal annotés par exemple. Il est donc essentiel de bien les définir avant de s'y intéresser précisément pour tenter d'en déduire des règles. De plus, il a été mis en évidence que les logiciels d'inférence de gènes sont leurrés par les pseudogènes qu'ils identifient pour certains comme des gènes potentiellement codants. Ainsi certaines mutations ou erreurs de séquençages peuvent induire des décalages de phase ou des stop en phase, induisant

la prédiction de deux objets géniques codants (CDS) au lieu d'un seul de statut pseudogénique comme observé dans caat-Box. Il est donc nécessaire de s'intéresser à ces séquences et de les analyser pour déterminer leur nature réelle et pouvoir poursuivre l'analyse avec un jeu de données plus robuste.

(i) Bactéries Lactiques

Définition Les « bactéries lactiques » tirent leur nom de leur principal mécanisme énergétique. En effet, elles sont dépourvues de chaîne respiratoire ce qui les rend incapables de générer de l'ATP par phosphorylation oxydative. Leur source d'énergie provient donc essentiellement de la phosphorylation de l'ADP en ATP au cours de la fermentation des sucres, dont le produit final majeur est l'acide lactique.

En plus de leur capacité fermentaire, les bactéries lactiques partagent des caractères morphologiques et physiologiques communs. La définition généralement admise regroupe ainsi « les bactéries en forme de coques ou bacilles, à Gram positif, immobiles, asporulées, anaérobies mais aérotolérantes, ne possédant pas de catalase, ni de nitrate réductase ou de cytochrome oxydase ». Cependant, il ne s'agit que d'une définition biologique et non d'une réelle classification taxonomique, même si les bactéries lactiques appartiennent pour la plupart à un même ordre.

Cet ordre taxonomique des *Lactobacillales* regroupe un grand nombre d'espèces réparties dans 6 familles et 35 genres différents (Ludwig et al., 2009) (voir Figure 4.70). Bien que la plupart de ces espèces répondent à la définition générale communément admise d'une bactérie lactique, seules quelques unes sont réellement considérées comme telles. En effet, dans le langage courant, le terme bactérie lactique fait uniquement référence aux espèces non pathogènes dont les propriétés fermentaires sont utilisées par l'homme. Les bactéries lactiques sont donc essentiellement associées aux genres : *Carnobacterium*, *Lactobacillus*, *Lactococcus*, *Leuconostoc*, *Oenococcus*, *Enterococcus*, *Pediococcus*, *Streptococcus*, *Tetragenococcus*, *Vagococcus* et *Weissella*.

Cette nuance permet également de distinguer des espèces très proches d'un point de vue phylogénétique, mais pourtant extrêmement différentes.

Données génomiques disponibles Les données relatives aux génomes des bactéries lactiques utilisées au cours de cette étude ont été extraites de la base de données GOLD [2] et de la base de données du NCBI [3]. Un jeu de 67 génomes bactériens entièrement séquencés et présentant un génome relativement compact a été sélectionné en fonction de l'appartenance taxonomique de chaque espèce.

2. `https://gold.jgi-psf.org/`
3. `http://www.ncbi.nlm.nih.gov/`

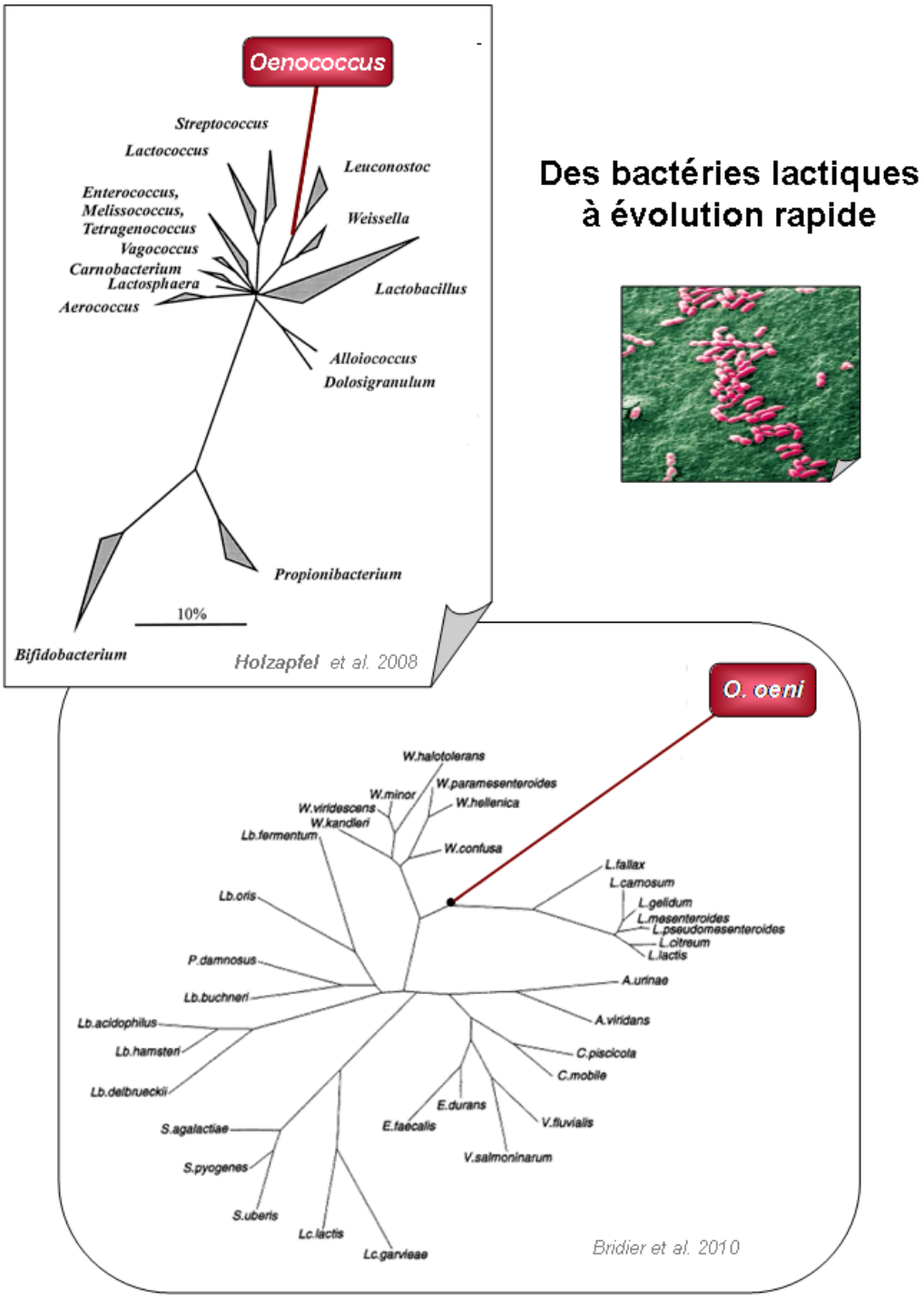


FIGURE 4.70 – *Arbre phylogénétique présentant l'embranchement du genre Oenococcus.*

(ii) Oenococcus oeni

Oenococcus oeni est une bactérie lactique en forme de coques (généralement en paire ou en chainette), acidophile, anaérobie facultative et hétérofermentaire. Elle fait partie de la flore bactérienne naturelle colonisant la surface de certains fruits, comme le raisin et la pomme, mais elle est surtout isolée dans le vin car elle y est l'espèce ultra-majoritaire pendant la fermentation malolactique. Grâce à une tolérance intrinsèque aux pH acides (pH 3,8-4,8) et aux forts degrés alcooliques (10-15% d'éthanol), elle est la bactérie qui supporte le mieux les conditions stringentes du vin après la fermentation alcoolique. Pour cette raison, le plus souvent c'est à cette espèce seule qu'incombe la responsabilité du déclenchement et du bon déroulement de la fermentation malolactique. *O. oeni* occupe ainsi une place importante en oenologie, d'autant que son utilisation comme principal levain malolactique tend désormais à la rendre incontournable.

Sur le plan taxonomique, l'espèce *O. oeni* se distingue également des autres bactéries lactiques. Considérée à l'origine comme un membre du genre *Leuconostoc*, elle a été nommée pendant longtemps *Leuconostoc oenos* (GARVIE, 1967). On a découvert par la suite de nombreuses différences entre cette espèce et les autres leuconostocs, notamment son caractère acidophile, son absence de glucose-6-phosphate deshydrogénase NAD-dépendante, ou encore le profil électrophorétique de ses protéines solubles (Garvie and Farrow, 1980; Garvie, 1986; Dicks et al., 1990). Les résultats des hybridations ADN-ADN et ARN-ADN, ainsi que le séquençage des ADN ribosomaux 16S et 23S (YANG and Woese, 1989) ; (Martinez-Murcia et al., 1993), ont également révélé des différences génétiques majeures entre *L. oenos* et tous les leuconostocs. En conséquence, *L.oenos* a été considéré à part et renommé *O. oeni*, première espèce du genre *Oenococcus* reconnu à cette occasion (Dicks et al., 1995).

Le séquençage de l'ADNr 16S et 23S a été déterminant pour le positionnement phylogénétique d'*O. oeni*. En effet, ces analyses ont confirmé son appartenance à la lignée *Leuconostoc-Weissella-Oenococcus*, mais elles ont surtout mis en évidence une très longue distance génétique séparant *O. oeni* des leuconostocs. Compte tenu également des nombreuses particularités biologiques de cette bactérie, (YANG and Woese, 1989) ont supposé qu'il s'agissait d'une espèce tachytélique, c'est-à-dire une espèce évoluant rapidement. Cette hypothèse tout d'abord remise en cause (Morse et al., 1996) ; (Chelo et al., 2007) est désormais soutenue fortement par la phylogénie établie à partir des premiers génomes de bactéries lactiques, comprenant celui de la souche *O. oeni* PSU-1 (Mills et al., 2005) (Makarova et al., 2006).

L'absence des gènes mutS et mutL (Makarova and Koonin, 2007), mise en évidence lors de l'étude du premier génome d'*O. oeni*, est une singularité génétique pouvant potentiellement expliquer la divergence rapide de cette espèce. Ils codent des enzymes clés du système MMR (MisMatch Repair), qui favorisent la conservation du génome en limitant les mutations et les recombinaisons (Eisen and Hanawalt,

1999). Plusieurs études, menées chez différentes espèces, rapportent en effet une augmentation de la fréquence des mutations et des transferts horizontaux lorsque les gènes mutS/L sont inactivés (Matic et al., 1995; Prunier and Leclercq, 2005)

En accord avec ce constat, une étude récente a révélé un taux de mutations spontanées (en présence d'antibiotiques) beaucoup plus élevé chez *O. oeni* (100-1000X) que chez les espèces proches *L. mesenteroides subsp. mesenteroides* et *Pediococcus pentosaceus* qui possèdent les gènes mutS/L (Marcobal et al., 2008). De manière intéressante, la même étude présente un résultat similaire (mais dans une moindre mesure, 10-100X) pour *Oenococcus kitaharae*, une bactérie récemment isolée du shochu (un alcool japonais à base de riz et de pomme de terre (Endo and Okada, 2006)). Cette bactérie est très différente d'*O. oeni* : elle n'est pas acidophile, ne pousse pas à 10% d'éthanol et ne réalise pas la fermentation malolactique. Elle a été reconnue comme la seconde espèce du genre *Oenococcus* principalement en raison de la séquence de son ADNr 16S et d'un pourcentage d'hybridation ADN-ADN avec *O. oeni* de 25-30 (Endo and Okada, 2006). Néanmoins, il semble que tous les isolats d'*O. kitaharae* et toutes les souches d'*O. oeni* aient en commun l'absence des gènes mutS/L, pourtant peu fréquente (Marcobal et al., 2008). Les auteurs suggèrent donc que la perte de ces gènes, par l'ancêtre commun aux deux espèces, a probablement entrainé l'apparition d'un phénotype hypermutateur qui aurait accéléré la divergence du genre *Oenococcus* (Marcobal et al., 2008) (voir Figure 4.70).

Compte-tenu de l'importance biotechnologique d'*O. oeni* pour l'industrie vinicole, un grand nombre de souches a depuis longtemps été isolé (on en décompte plus de 200 à ce jour) et caractérisé, en particulier par les fabriquants de levains malolactiques. L'expérience, tant chez les producteurs de vin que de levain, montre qu'il existe une grande diversité au sein de cette espèce. Les souches d'*O. oeni* possèdent naturellement des propriétés variables qui les rendent plus ou moins adaptées à un usage oenologique. La plus discriminante est leur capacité à réaliser la fermentation malolactique, qui dépend directement de leur aptitude à survivre et à proliférer dans le vin. Les microvinifications pratiquées en laboratoire ont démontré que certaines souches poussent rapidement dans différents vins et différentes conditions (pH, % d'éthanol, ...) et réalisent la dégradation complète du malate. Tandis que d'autres sont beaucoup plus lentes, voire ne survivent pas après l'inoculation (Martineau and Henick-Kling, 1995).

En dehors de leur capacité a assurer la fermentation malolactique, les souches d'*O. oeni* présentent d'autres phénotypes variables qui sont moins décisifs, mais tout de même importants pour la vinification. Elles ont, par exemple, des activités glycosidases très différentes, à la fois en terme d'efficacité et de nature de substrat (Grimaldi et al., 2005). Ce paramètre peut avoir une influence sur la qualité organoleptique du vin car ces réactions enzymatiques libèrent des composés aromatiques du raisin ou du bois de chêne présents sous forme glycosylée (Boido et al., 2002). De manière générale, plusieurs études indiquent des variations perceptibles de l'arôme du vin en fonction de la souche d'*O. oeni* utilisée pour réaliser la fermentation malolactique (Henick-Kling et al., 1994). D'autre part, certaines altérations du vin sont

également associées à la présence de souches spécifiques. Il existe ainsi une diversité phénotypique avec des souches à fortes performances biotechnologiques ou à performances biotechnologiques plus modestes.

Données génomiques disponibles Actuellement, le genre *Oenococcus* presente deux espèces *O. oeni* et *O. kitaharae*. L'espèce la mieux connue est l'espèce *O. oeni* caractérisée par une organisation complexe constituée de 258 souches organisées en phylogroupes distincts, mettant en évidence une diversité génétique certaine. À l'heure actuelle, seuls les génomes de deux souches d'*O. oeni* ont été séquencés, assemblés et annotés, il s'agit des souches PSU-1 (public : NC_008528) et BAA-1163 (version publique partielle : NZ_AAUV00000000 ; version complète : non publique) qui présentent des performances oenologiques opposées.

Pour cette étude nous avons analysé les génomes complets des souches PSU-1 (NC_008528 ; (Mills et al., 2005)) et ATCC BAA-1163 (WGS35 assembly). Ces souches appartiennent à deux sous groupes phylogénétiques distincts et présentent des capacités biotechnologiques opposées (Borneman et al., 2010; Bartowsky and Borneman, 2011). La première annotation de leur chromosome complet (1.8 Mb, 38%-GC) a permis d'inventorier 1691 and 1674 gènes codant pour des protéines ainsi qu'un ensemble de, respectivement, 122 et 155 pseudogènes hypothétiques (Mills et al., 2005; Bon et al., 2008).

4.3 *PseudOE* : Une Procédure d'Identification des Pseudogènes

Afin d'établir un inventaire exhaustif des pseudogènes potentiels, nous avons développé un ensemble de procédures semi-automatiques distribuées en quatre modules. Cette méthode, PseudOE, permet la détection des pseudogènes issus de mutations ponctuelles ou de troncations, mais pas de l'altération de ses séquences régulatrices.

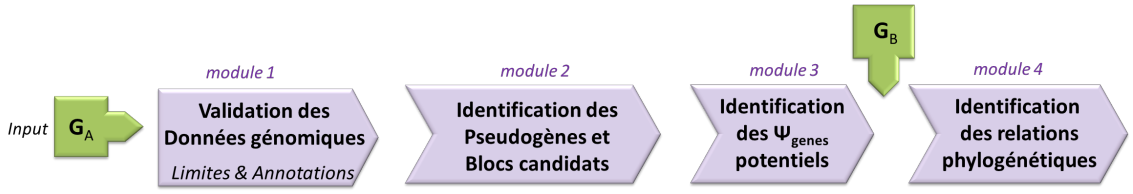


Figure 4.71 – *Étapes de l'inférence des séquences pseudogéniques au sein d'un génome* ***GA****, puis comparaison avec un génome* ***GB****.*

4.3.1 Consolidation des Données d'Origine

Les génomes séquencés n'ont pas tous été annotés ou du moins précisément annotés : dans la plupart des cas seuls leurs CDS et ARN classiques ont été répertoriés. Il est donc indispensable de reprendre au préalable ce répertoire génique initial des souches sélectionnées pour les vérifier, les consolider et le cas échéant les corriger.

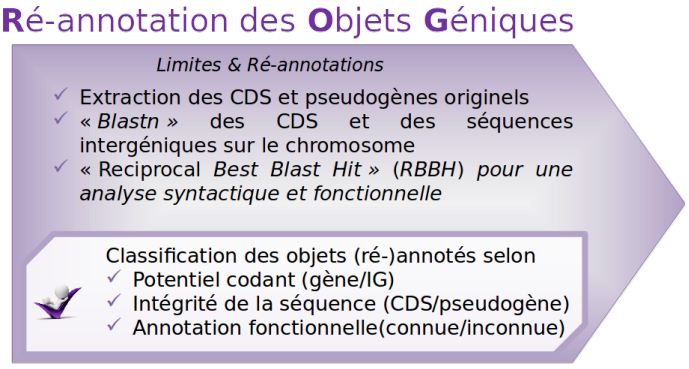


FIGURE 4.72 – *Étape de ré-annotation des séquences géniques.*

Étude des coordonnées des séquences Au cours de cette première étape, le fichier GenBank original est utilisé comme matrice afin d'extraire les séquences des différents objets géniques codants ou non codants, intacts ou altérés, précédemment identifiées sur le chromosome. Toutes les séquences codantes, non codantes ou intergéniques sont comparées, à l'aide du filtre BLAST 2.3.2 (Altschul et al., 1997), avec leur séquence chromosomique. BLAST est utilisé sans filtrage des régions de faible complexité. Afin de sélectionner les séquences potentiellement pseudogéniques les critères de sélection suivant sont appliqués :

- $e - value \leq 10^{-30}$
- similarité$\geq 50\%$
- couverture de l'alignement$\geq 70\%$

Les alignements obtenus permettent à la fois de valider les séquences initialement définies et d'identifier de potentielles séquences pseudogéniques. Dans ce dernier cas, les séquences identifiées sont étendues de part et d'autre jusqu'aux codons start et stop. Cette procédure contribue ainsi à la redéfinition des gènes codant pour des protéines et à l'enrichissement du répertoire génique initial en déterminant de possibles nouvelles séquences. À la fin de cette première étape une nouvelle liste des objets géniques de trois répertoires distincts : CDS (CDS), pseudogènes (ψ) et intergènes (IG), est définie.

Ré-annotation des séquences La procédure du « Reciprocal Best Blast Hit » est appliquée sur les pseudogènes candidats obtenus au cours de l'étape précédente. Pour cela, un BLASTx (avec une matrice de type BLOSUM 62 et sans filtrage des régions

de faible complexité) est réalisé contre la base de données protéique non redondante Genbank du NCBI. Suite à une première sélection automatique des résultats de BLAST ($e - value \leq 10^{-30}$; similarité$\geq$ 50% ; couverture de l'alignement$\geq$ 70%), la séquence menant au meilleur alignement est sélectionnée comme étant un homologue potentiel. Les protéines ainsi sélectionnées sont alors utilisées comme appât pour réaliser le « reciprocal tBLASTn » contre la séquence nucléotidique du chromosome d'origine. Si les séquences alors identifiées ne sont pas alignées au niveau de la séquence chromosomique originelle, celles-ci sont écartées du reste de l'analyse. Si une séquence présente deux séquences correspondantes à proximité l'une de l'autre sur le chromosome, ces séquences sont fusionnées. En effet, ces deux séquences pourraient correspondre aux deux portions d'une unique séquence. Au contraire, si les deux séquences correspondantes sont assez éloignées, elles pourraient provenir d'un événement de fission d'une séquence par l'insertion d'un îlot génomique.

Une telle stratégie est nécessaire à l'annotation syntaxique et fonctionnelle des différents répertoires géniques. Cela permet, dans un premier temps, d'« exhumer », d'extraire et d'identifier les pseudogènes candidats par alignement avec leur séquence native et, dans un second temps, d'inférer pour chacune de ces séquences ses coordonnées sur le chromosome et une fonction possible. Au cours de cette étape on discrimine les séquences géniques (CDS_{temp} ou ψ_{temp}) des séquences intergéniques (IG_{temp}). Les séquences intergéniques sont étiquetées comme contenant un pseudogène candidat si une séquence codant pour une protéine est alignée, au moins en partie, sur la région intergénique. Les objets chromosomiques d'origine et ceux redéfinis sont alors classés selon leur statut potentiel (intergène ou gène), l'intégrité de leur séquence (CDS ou pseudogène) et le statut de leur annotation fonctionnelle (prédit ou indéfini).

4.3.2 Identification des Blocs de Séquences Candidates

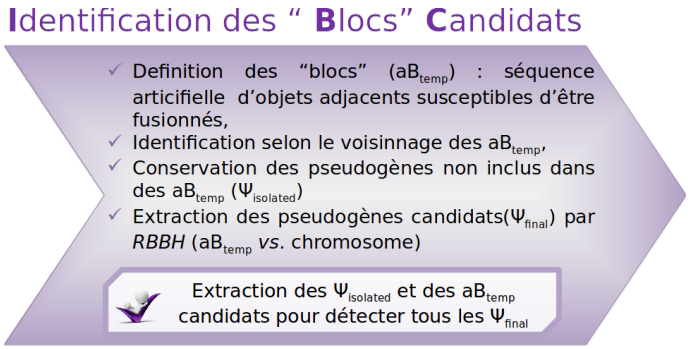


Figure 4.73 – *Étape d'identification des blocs de séquences.*

Les objets chromosomiques, gènes ($\sum \text{CDS}_{temp}$) et pseudogènes potentiels ($\sum \psi_{temp}$) identifiés, sont analysés afin d'en extraire des « blocs » (aB_{temp})sur la

séquence chromosomique du génome analysé. Ces entités artificielles et temporaires sont définies comme une série de séquences adjacentes qui pourraient présenter des coordonnées initiales et terminales erronées ou des caractéristiques qui justifieraient leur fusion en une seule et unique séquence génique et donc de les reclasser comme des séquences pseudogéniques potentielles. Le statut des séquences adjacentes, à savoir colinéaires ou non, de chaque membre d'un aB_{temp} ainsi que la congruence des annotations de chacun des membres (fonction similaire ou non), ont été utilisés comme critères de définition de ces aB_{temp}. Ainsi tant que deux séquences voisines présentent une annotation similaire l'aB_{temp} est étendu. Les extrémités de l'aB_{temp} sont alors données par le début de la première séquence et la fin de la dernière séquence qui le composent plus ou moins 100bp. Les séquences des pseudogènes non compris dans un aB_{temp}, ou pseudogènes hypothétiques isolés ($\psi_{isolated}$), et les aB_{temp} identifiés constituent le jeu de données qui va être analysé par le reste du pipeline.

4.3.3 Identification des Pseudogènes Potentiels

La méthode du « reciprocal best blast hit » est à nouveau employée au cours de cette étape afin d'établir le catalogue final des pseudogènes candidats. Une phase d'alignement avec BLASTx des séquences identifiées au cours de l'étape précédente, à savoir les $\psi_{isolated}$ et les aB_{temp}, est réalisée contre la base de données protéiques généraliste non redondante Genbank. Les alignements sont alors évalués afin de sélectionner les séquences correspondant aux meilleurs alignements.

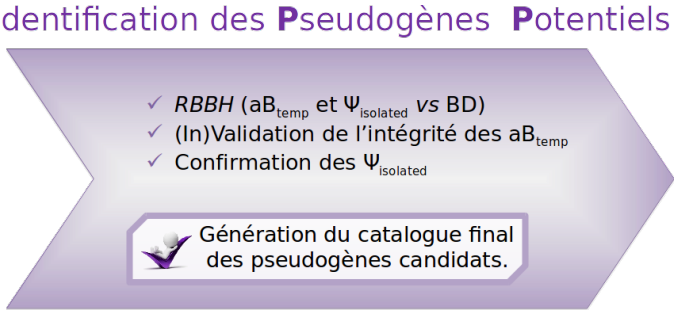


FIGURE 4.74 – *Étape d'identification des séquences pseudogéniques potentielles.*

La géométrie des alignements calculés permet d'évaluer les limites de la séquence en entrée (Q) et de retenir le meilleur hit (S). On distingue deux situations :

- Si seule une séquence S s'aligne sur une longueur supérieure ou égale à 70% de la séquence en entrée alors cette séquence est retenue comme la meilleure séquence alignée et la séquence en entrée est considérée comme un pseudogène hypothétique avec l'annotation de la séquence alignée retenue.
- Si deux séquences au moins s'alignent sur la séquence en entrée et ce de manière non chevauchante, alors la séquence alignée la plus à gauche, notée S_l, et celle la plus à droite, notée S_r, sont utilisées comme des ancres afin de

déterminer la nature de la séquence en entrée. On distingue à nouveau deux cas :

- Si S_l et S_r constituent les deux extrémités d'une même séquence S et que les séquences comprises entre les deux sont codantes ou non codantes, alors la séquence en entrée est comptabilisée comme un pseudogène potentiel (suite à un événement de fission) et S est retenue comme la meilleure séquence alignée.
- Si S_l et S_r ne constituent pas les deux extrémités d'une même séquence, alors, que les séquences comprises entre les deux extrémités soient codantes ou non, le bloc est divisé et toutes les meilleures séquences alignées non chevauchantes sont retenues.

Le jeu des séquences protéiques retenues est utilisé comme appât et chaque séquence (Q) est alors alignée sur la séquence nucléotidique (S) du chromosome par tBLASTx. Les mêmes critères de sélection que pour le premier Best Blast Hit sont appliqués pour identifier les pseudogènes potentiels. On distingue à cette étape plusieurs cas :

- Si la séquence s'aligne sur une CDS, alors c'est une CDS présentant probablement un paralogue altéré, on oublie donc ses coordonnées.
- Si la séquence s'aligne sur une séquence intergénique, alors la séquence chromosomique alignée est un pseudogène issu d'une mutation non sens, ψ_{stop}, ou de multiples mutations invalidantes, $\psi_{erosion}$, ou d'une troncature si les partie N-term ou C-term codant pour une protéine sont absentes, ψ_{tronc}.
- Si la séquence s'aligne sur une séquence intergénique et une CDS adjacente, alors la séquence alignée est un ψ_{stop} ou si la séquence présente un motif de décalage de phase, ψ_{FS}.
- Si la séquence s'aligne sur deux CDS adjacentes, alors la séquence alignée est un ψ_{stop} ou si la séquence présente un motif de décalage de phase, ψ_{FS}.
- Si la séquence s'aligne sur deux portions non adjacentes du chromosome, alors la séquence est issue d'un événement de fission, $\psi_{fission}$. La séquence pseudogénique est alors artificielle puisque divisée dans le génome.

L'ensemble des pseudogènes recensés constitue alors le répertoire pseudogénique final. Ce dernier combiné aux répertoires originaux permet par approche différentielle d'établir également le répertoire final des CDS et des séquences intergéniques.

4.3.4 Relations Phylogénétiques

Ce dernier module de la méthode *PseudOE* permet d'intégrer une analyse phylogénétique de base du répertoire pseudogénique recensé. Il permet d'évaluer, pour un pseudogène donné la présence ou l'absence de copies intra ou inter-génomiques et :

- sa redondance au sein d'un génome donné.
- sa conservation au cours de l'évolution du point de vue intra ou inter-spécifique.

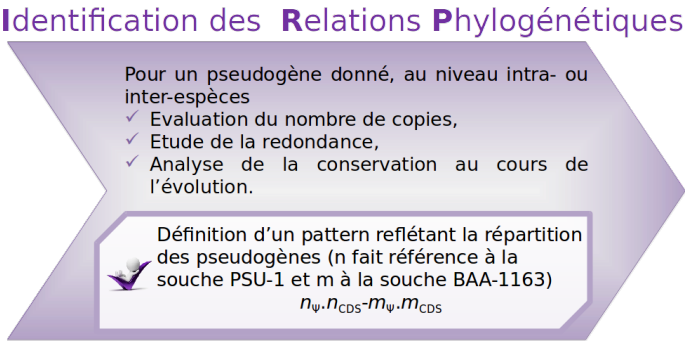


FIGURE 4.75 – *Étape de caractérisation des relations phylogénétiques entre séquences géniques.*

Comme les pseudogènes sont des copies altérées de gènes connus, cette analyse est réalisée avec un « reciprocal best tBLASTn hit ». On identifie alors des paires homogènes de pseudogènes homologues, $homo-\psi-paire$, et des paires hétérogènes de pseudogènes et CDS, $hetero-\psi-paire$, qu'il est possible d'identifier respectivement au sein d'un génome ou entre deux génomes.

Le répertoire de pseudogènes final déterminé dans un génome Q (ψ_Q) est aligné par tBLASTn avec la séquence chromosomique du génome lui-même pour identifier les hypothétiques gènes paralogues (G_Q), puis aligné par tBLASTn avec un deuxième génome T afin d'identifier les possibles gènes orthologues (qu'ils soient intacts ou altérés).

Les alignements présentant une $e-value \leq 10^{-30}$, un minimum de 50% de similarité en séquence et une couverture supérieure ou égale à 70% sont alors conservés afin d'identifier quel type de relation relie les deux séquences. En ne tenant pas compte de l'alignement du pseudogène avec sa propre séquence sur son chromosome on détermine :

- sur son chromosome : l'ensemble des séquences intactes et altérées paralogues, $CDS_{paralogs}$ et $\psi_{paralogs}$.
- sur le chromosome d'une autre souche : l'ensemble des séquences intactes et altérées orthologues, $CDS_{orthologs}$ et $\psi_{orthologs}$.

Ces critères de hiérarchisation permettent de discriminer les pseudogènes unitaires ($\psi_{isolated}$), sans paralogues ni orthologues, des pseudogènes faisant partie d'une famille multigénique.

Le degré de redondance d'un pseudogène au sein d'un génome est donné par un pattern élaboré pour ce module. Deux versions de ce pattern sont générées, la première, brute donne accès directement au nombre de copies identifiées, la seconde, lissée, donne uniquement des informations sur la présence ou l'absence de ces copies. Ce pattern est formalisé comme suit : $n_{\psi}.n_CDS-m_{\psi}.m_CDS$, où les variables n et m indiquent respectivement le nombre de copies (pseudogènes puis CDS, séparés par un point) dans le génome d'origine et dans le génome comparé, séparé par un tiret (voir Figure 4.76). Le pattern de présence/absence est alors obtenu en ramenant à 1 toutes les valeurs supérieures à 0.

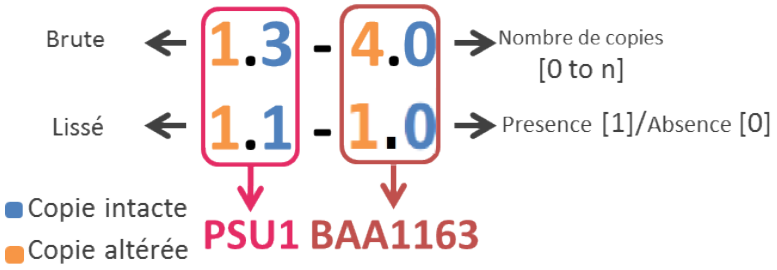


Figure 4.76 – *Pattern de comparaison de la composition pseudogénique de deux souches.*

4.3.5 Performances de *PseudOE*

Comparaison avec les outils d'identification de pseudogènes Quelques outils permettant l'identification des pseudogènes ont été développés à ce jour mais la plupart ciblent les organismes eucaryotes (*cf.* Section 1.3.2.(vi)). Néanmoins, l'un d'eux, s'intéressant aux pseudogènes bactériens, a retenu notre attention. Il s'agit de la suite logiciel *Psi-Phi* (Lerat and Ochman, 2004) qui se décompose en deux modules perl : le premier qui traite les données issues d'un alignement de séquences protéiques entre le génome d'intérêt et un génome de référence et sélectionne les séquences génomiques les plus semblables ; le second qui traite les sorties du module précédant ne permet de prédire qu'un faible nombre de pseudogènes. Il est important de noter que les pseudogènes sont ici prédits à partir d'un seul génome proche, peu de pseudogènes seront détectés et la prédiction n'est pas exhaustive. Le code de cet outil est disponible sur simple demande aux auteurs. Il apparait intéressant de comparer les résultats fournis par cet outil avec ceux trouvés à l'issue de l'analyse avec *PseudOE*.

Les génomes de PSU-1 et BAA-1163 dans leurs versions originales ont été analysés avec le programme *Psi-Phi* en utilisant la souche PSU-1 comme référence pour étudier la souche BAA-1163. L'analyse comparative de *Psi-Phi* a permis la détection de 96 pseudogènes potentiels également prédits par la méthode *PseudOE*. 86 d'entre eux ont été prédits avec les même coordonnées et 10 présentent de faibles variations dans ces coordonnées. On observe aussi que 49 candidats proposés par *PseudOE* ne le sont pas par *Psi-Phi*. On obtient un taux de sous prédiction de *Psi-Phi* de $33,8\%$ et un taux de sur-prédiction de 0% de *Psi-Phi* par rapport à *PseudOE*. Ces différences peuvent être expliquées par le fait que *Psi-Phi* se limite à l'identification de CDS tronquées (Lerat and Ochman, 2004). Mais aussi par la méthode comparative employée par *Psi-Phi* qui se limite donc à la comparaison avec le génome dit de référence.

4.4 Analyse Comparative et Topologique de la Pseudogénisation

4.4.1 Inventaire des Pseudogènes

Dans chacun des génomes étudiés, la méthode *PseudOE* a permis la mise en évidence d'un nombre substantiel de potentiels nouveaux gènes altérés provenant d'un décalage de phase de lecture ou bien de délétions plus ou moins importantes au sein de la CDS originale. Pour cela, les objets géniques altérés recensés dans les génomes ont subi trois niveaux d'annotation graduelle :

- une **annotation syntactique** qui a permis de définir des populations pseudogéniques de types gènes mutés (avec ou sans décalage de phase) et de type gènes fragmentés (tronqués, fissurés, errodés).
- une **annotation fonctionnelle** a été par la suite réalisée par inférence à partir des meilleurs homologues identifiés par les recherches de séquences similaires dans les banques de données généralistes et spécialisées. La stratégie d'annotation fonctionnelle retenue pour cette étude consiste en l'alignement des pseudogènes candidats avec les séquences contenues dans les banques **NR** et **NT**.
- une **annotation relationnelle** est également entreprise. Il s'agit dans un premier temps de recenser la présence de copies plus ou moins parfaites au sein d'un même génome des pseudogènes, et d'établir des liens éventuels de paralogie unissant les gènes d'une même famille. Il s'agit dans un second temps, de dresser une cartographie physique de la localisation géographique précise de chaque objet génique à l'aide du logiciel CIRCOS. Dans un troisième et dernier temps il s'agit d'évaluer le degré de conservation phylogénétique à l'échelle intraspécifique (micro-évolution) et à l'échelle interspécifique (macro-évolution). Pour ce faire l'algorithme d'alignement de séquences doit de nouveau être employé ici mais de manière différente. En effet, les pseudogènes d'une souche A sont alignés contre eux-même mais aussi contre les CDS et les intergènes de la souche A puis contre les pseudogènes et les CDS de la souche B, et réciproquement. Ce sont ici des lots de séquences qui doivent être comparés entre eux séquence par séquence. Ce qui doit permettre d'identifier pour les séquences pseudogéniques :
  - la redondance (ou présence de paralogues) : redondance active ou inactive
  - la spécificité à une souche : présence ou absence de paralogues et d'orthologues, « actifs » ou « inactifs »
  - la conservation phylogénétique :
    - dans l'espèce *O. oeni* : entre PSU-1 et BAA-1163
    - dans le genre *Oenococcus* : avec *O. kitaharae*
    - dans la famille des *Leuconostocs* : avec *Leuconostoc mesenteroides* ou *Leuconostoc citrum*

- dans l'embranchement des *Lactobacillales* : avec *Lactibacillus plantarum*
- le positionnement de ces pseudogènes sur le génome, soit leur cartographie

En parallèle, cela a diminué le nombre de CDS précédement prédites dans PSU-1 (soit 1634, $-3,4\%$) et BAA-1163 (soit 1570, $-6,2\%$) et augmenté le nombre de pseudogènes dans PSU-1 (145, $+18,9\%$) alors que dans BAA-1163 ce nombre reste presque inchangé (153, $-1,3\%$) (voir la Table 4.4).

Souche	Chr. (bp)	Version	Genes	ARN classiques	CDS	ψ	Total Gènes	Gènes Intacts	Gènes Altérés	Potentiel Codant
PSU-1	1780517	Avant	1864	51	1691	122	1813	93.3%	6.7%	82.5%
		Après	1830	51	1634	145	1779	91.8%	8.1%	80.5%
BAA-1163	1792086	Avant	1880	51	1674	155	1829	91.5%	8.5%	84.4%
		Après	1774	51	1570	153	1723	91.1%	8.9%	81.2%

Table 4.4 – *Comparaisons des données sur les génomes PSU-1 et BAA-1163 avant et après l'utilisation de PseudOE.*

Les répertoires nouvellement redéfinis entrainent une faible variation ($-3,2\%$) du nombre potentiel de gènes codant pour des protéines, qui demeurent néanmoins les constituants les plus présents de ces génomes (en moyenne $\sim 81\%$) en comparaison avec les autres génomes bactériens. Ce qui souligne le fort taux de compaction des génomes d'*O.oeni* sous la pression d'adaptation à son environnement le vin. On note que ce potentiel codant est très similaire entre les deux souches d'*O. oeni* étudiées avec une différence de seulement $0,7\%$. Néanmoins, une variation de cette métrique pouvant atteindre 10% a été enregistrée entre deux souches d'*O. oeni*. Ce qui s'explique par l'étendue des variations intra-spécifiques (Borneman et al., 2010; Bon et al., 2009).

4.4.2 Populations Pseudogéniques : Plasticité Génique

L'analyse approfondie des données issues de l'analyse avec PseudOE a permis d'émettre certaines hypothèses quant à l'existence de règles (spécifiques à l'espèce ou à la souche) régissant l'altération dynamique des gènes dans les génomes d'*O. oeni*.

Les objets géniques candidats ont été analysés manuellement et l'ensemble des événements d'altération ont été répertoriés (mutation, troncature, fission,...) et classés afin d'établir le spectre complet des événements à l'origine de la génération de pseudogènes. Pour cela, *T-COFFEE* (Taly et al., 2011) a été utilisé en alignant les séquences nucléotidiques des pseudogènes candidats avec leurs séquences fonctionnelles associées.

(i) Analyse Syntactique

Événement \ Souche	PSU-1	BAA-1163
Codon STOP	36%	39%
Frameshift	26%	37%
Fragmentation	12%	7%
Érosion	23%	14%
Fission/Insertion	3%	4%
TOTAL	145ψ	153ψ

TABLE 4.5 – *Classement des pseudogènes selon l'événement d'altération à l'origine de leur état non codant dans les deux génomes d'intérêt.*

La plupart des pseudogènes semblent résulter de la dégradation de gènes natifs fonctionnels. La mutation de CDS et l'érosion sont de loin les deux mécanismes moléculaires liés à l'évolution prédominant dans l'altération des gènes. Les décalages de phase entre deux CDS chevauchantes consécutives semblent tenir un rôle non négligeable dans la plasticité des génomes d'*O. oeni*. Même si elles sont moins fréquentes, les fragmentations de tailles variables, et les fissions, sont également impliquées dans l'organisation des génomes d'*O. oeni*. Pas moins de 11 transferts horizontaux de gènes provenant d'éléments transposables ou de l'intégration de phage, ont été identifiés dans des gènes interrompus dans PSU-1 (avec une taille maximale de 24kb) et dans BAA-1163 (avec une taille maximale de 13kb). Malgré cette tendance générale, les frameshift et les gènes fragmentés apparaissent plus ou moins fréquemment dans BAA-1163 comparé à PSU-1 (voir Table 4.5). De telles différences tendent à expliquer l'impact de la pression de la niche écologique sur la génération de variations intra-espèce, et donc sur la plasticité des génomes '*O. oeni* (Bon et al., 2009; Bartowsky and Borneman, 2011).

(ii) Analyse Fonctionnelle

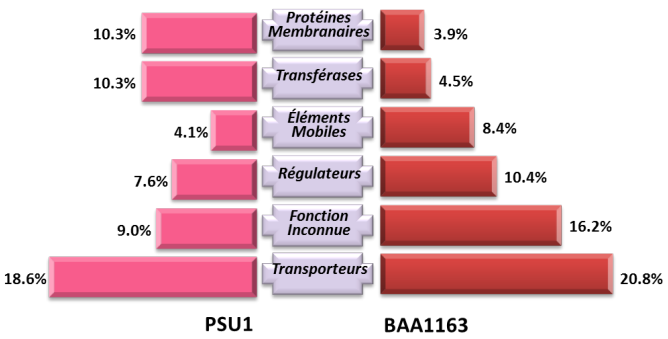
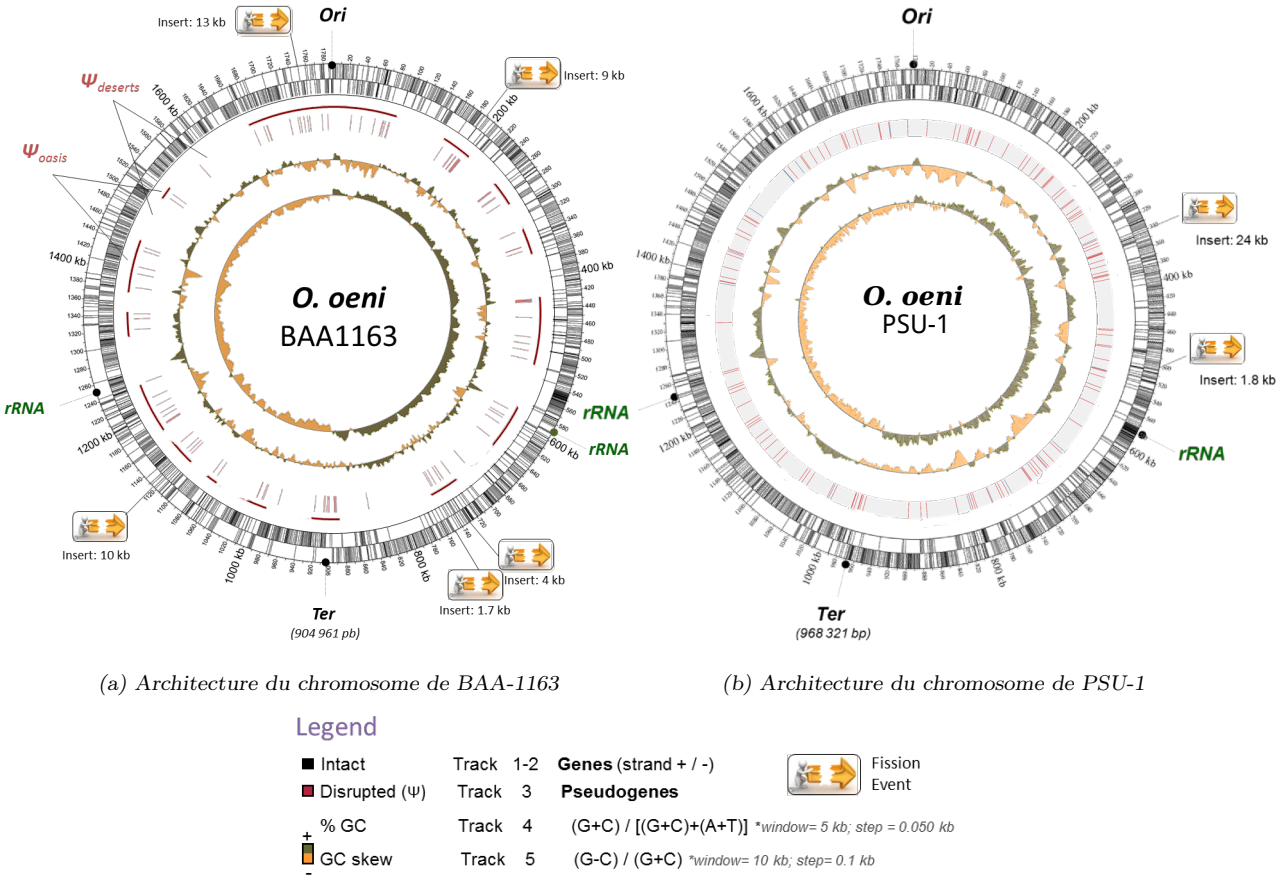


FIGURE 4.77 – *Tonalité générale de la prolifération des pseudogènes au sein des différentes fonctions génomiques.*

On observe que près de $\sim 80\%$ des pseudogènes peuvent être associés à une fonction biologique cellulaire. Certaines des ces fonctions semblent même plus sujettes à la pseudogénisation, comme les transporteurs par exemple (voir Figure 4.77). La prédominance de la pseudogénisation observée au sein de certaines fonctions pourrait avoir un rôle important de marqueur de fonctions « dispensables ».

La spécificité de souche des fonctions pseudogénisées, à savoir les transférases et les protéines membranaires chez PSU-1 et les régulateurs transcriptionnels et les éléments mobiles chez BAA-1163, suggèrent que les pseudogènes pourraient être des marqueurs génétiques des différences phénotypiques entre souches. Ils pourraient ainsi moduler l'adaptation des souches à leur environnement, à savoir le vin dans cette étude.

(iii) Analyse Relationnelle



(a) Architecture du chromosome de BAA-1163 *(b) Architecture du chromosome de PSU-1*

Figure 4.78 – *Topologie de la répartition des pseudogènes le long du chromosome bactérien.*

Distribution topologique non aléatoire Afin de visualiser la distribution géographique des pseudogènes candidats prédits le long du chromosome bactérien et d'identifier une éventuelle distribution organisée de ces objets, nous avons utilisé

l'outil *CIRCOS* (Krzywinski et al., 2009).

L'analyse de la distribution géographique des pseudogènes le long du chromosome met en avant un schéma organisationnel spatial complexe et inattendu. L'étude des données préliminaires indique qu'outre des événements locaux, les pseudogènes seraient plutôt localisés sur des territoires chromosomiques définis non aléatoirement. Ce qui soutient le modèle mettant en avant des territoires ψ_{oasis} alternant avec des territoires ψ_{desert} le long du chromosome. Ce modèle fait écho aux récents travaux sur la répartition des séquences géniques selon le repliement de la molécule d'ADN (Junier et al., 2010; Mathelier and Carbone, 2010). En effet, ces articles mettent en évidence une architecture des territoires géniques synchronisée avec le repliement de l'ADN et la présence de territoires riches en séquences géniques.

Redondance au sein d'un génome : expansion génique On nomme $\psi_{paralogues}$ les gènes paralogues (Koonin, 2005), qu'ils soient intacts ou non, à une séquence pseudogénique. Seuls $\sim 4,9\%$ des pseudogènes de PSU-1, ψ_{PSU-1}, et $\sim 3,9\%$ des pseudogènes de BAA-1163, $\psi_{BAA-1163}$, présentent des copies extra-numéraires (n=3 au maximum, voir Table 4.6) et peuvent donc être regroupés au sein de familles multigéniques. Dans ce cas, ces copies proviennent de phénomènes de duplications indépendants puisque aucune duplication génome complet n'a eu lieu dans cette espèce. Quelle que soit la souche étudiée, les pseudogènes présentant un paralogue intact fonctionnel (polymorphisme des pseudogènes) sont rares et représentent seulement $\sim 2,7\%$ de la totalité du répertoire pseudogénique. L'ensemble de ces valeurs tend à prouver que, chez *O. oeni*, les pseudogènes proviennent préférentiellement de familles monogéniques.

4.4.3 Plasticité du Pseudome et Évolution

(i) Prévalence des Pseudogènes chez les bactéries lactiques

La distribution phylogénétique des fréquences génomiques des pseudogènes, calculées comme le ratio du nombre de pseudogènes sur le nombre de séquences géniques (codant pour des protéines et pseudogéniques), a été analysée au niveau intra- et inter-spécifique (voir Figure 4.79).

Au sein des génomes bactériens sélectionnés (*cf.* Section (i)), les pseudogènes sont détectés dans de faibles proportions, avec une fréquence comprise entre 1% et $\sim 5\%$ (avec une médiane à $\sim 1,4\%$). Avec une proportion de $\sim 8,5\%$ pour un génome de $1,8Mb$, soit ~ 150 pseudogènes, les génomes d'*O. oeni* apparaissent comme des réservoirs à pseudogènes en comparaison avec les autres génomes de bactéries lactiques (dont la médiane est à $\sim 2,3\%$). En revanche, les génomes de *L. mesenteroides* et *L. citrum*, étroitement liés aux *Oenococcus*, comptent respectivement 1970 et 1702 gènes intacts codant pour des protéines mais seulement 19 et 1 pseudogènes. Cette étude suggère que les génomes d'*O. oeni* pourraient constituer un exemple de génome érodé.

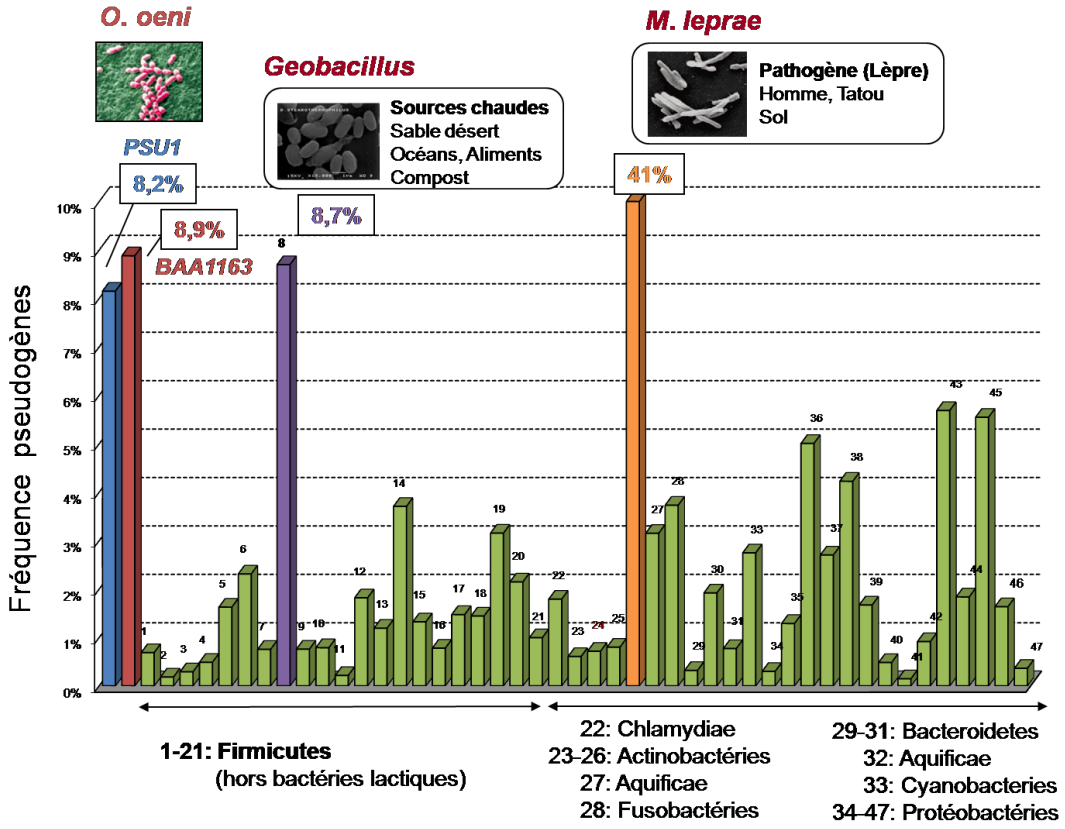


Figure 4.79 – *Distribution phylogénétique au sein de diverses souches bactériennes des pseudogènes recensés dans GOLD.*

À l'instar de *Geobacillus* et de certains pathogènes, *O. oeni* est un cas particulier de part sa forte concentration en pseudogènes. Ceci pourrait traduire une stratégie particulière d'adaptation aux environnements extrêmes que constituent le vin ou encore les sources chaudes.

(ii) Règles Phylogénétiques propres à *O. oeni*

La détermination du taux d'extinction des gènes, c'est-à-dire d'altérations ou d'absence, est importante afin d'élucider l'historique des événements survenus au cours de l'évolution (voir Figure 4.80) et menant à la réduction du génome en réponse à son adaptation à une niche écologique particulière.

Core-pseudome et Accessory-pseudome : propagation génique Les 298 pseudogènes ont été utilisés comme base pour retracer leur évolution et leur répartition au sein des souches d'*O. oeni* PSU-1 et BAA-1163 (voir Table 4.6). La plus grande partie des pseudogènes ($\sim 64\%$) est spécifique à une seule des deux souches.

Pour une souche donnée, environ $\sim 68\%$ des $\psi_{specifique}$-gènes présentent un orthologue intact dans la seconde souche mais pas de paralogue intact dans leur propre séquence génomique. De même, environ 30% de ces $\psi_{specifique}$-gènes sont des pseudogènes unitaires, c'est-à-dire des pseudogènes qui ne présentent ni copie intacte ni copie altérée dans son génome mais aussi dans le second génome. L'ensemble de ces données indique que les souches d'*O. oeni* sont enclins à l'accumulation des pseudogènes de manière indépendante. Ces données illustrent également le fait que l'altération des gènes est un processus dynamique au sein de cette espèce. Une telle évolution peut être expliquée par l'absence du système de réparation des mésappariements. Cette absence de correction est à l'origine d'un fort taux de mutations et donc d'une accumulation des erreurs spontanées au cours de la réplication de l'ADN (Marcobal et al., 2008).

Pseudome	**Conservation**	**Pattern Lissé**	**Nombre**	**Pattern Brut**	**Nombre**
Core	Common	1.0-1.0	49	1.0-1.0	47
				1.0-2.0	1
				2.0-1.0	1
Accessory	PSU-1 spécifique	1.0-0.1	65	1.0-0.1	65
		1.0-0.0	24	1.0-0.0	24
		1.1-0.1	2	1.1-0.1	1
				1.2-0.3	1
	BAA-1163 spécifique	0.1-1.0	64	0.1-1.0	64
		0.0-1.0	33	0.0-1.0	33
		0.1-1.1	2	0.1-1.1	2

TABLE *4.6 – Architecture du pseudome déduite de l'analyse comparative de PSU-1 et BAA-1163. Le pattern phylogénétique reflète l'expansion intra-génomique et inter-génomique des ψ gènes entre PSU-1 et BAA-1163.*

Cela peut également expliquer la facilité de génération de pseudogènes unitaires présentant des mutations bénéfiques et permettant une meilleure adaptation à leur environnement. Cette analyse donne une première vision de l'architecture du pseudome qui se compose d'un jeu, de petite taille, de pseudogènes conservés et probablement hérités de l'ancêtre commun aux deux souches, et un jeu de données plus large de pseudogènes spécifiques à chaque souches et donc d'origine plus récente. Ces deux jeux de données constituent respectivement le core-pseudome et l'accessory-pseudome.

Conclusion & Perspectives

Les pseudogènes constituent des leurres linguistiques et fonctionnels (via leur ARN) qui impactent la construction des familles géniques, la reconstruction des génomes ancestraux et l'expression de certains gènes.

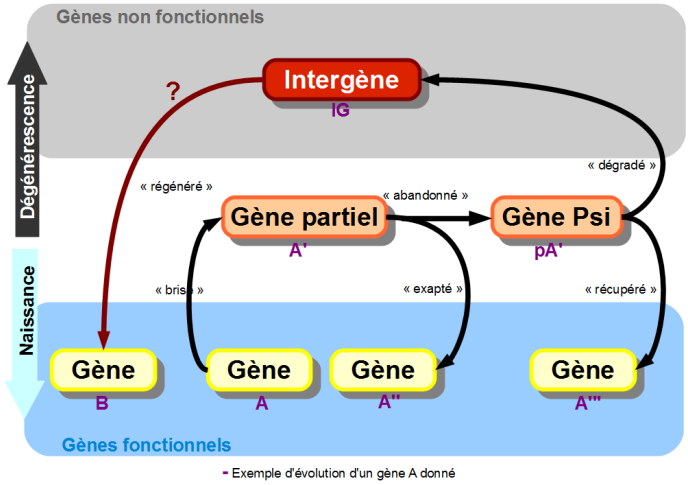


Figure 4.80 – *Cycle des événements évolutifs d'une séquence génique.*

La méthode *PseudOE* a permis la détection de nouveaux pseudogènes candidats au sein des régions non annotées du génome mais également de redéfinir les répertoires géniques potentiels. Les données collectées suite à l'utilisation de *PseudOE* ont permis une analyse plus avancée des génomes étudiés. Ces données ont alors servi de support à une étude comparative de la répartition des pseudogènes potentiels et de l'architecture de ces populations au sein des génomes d'*Oenococcus*, à savoir des génomes de bactéries lactiques hautement spécialisés et adaptés à des environnements stressants. Grâce à ces analyses, il est possible d'appréhender les mécanismes à l'origine de la plasticité génique des génomes d'*O. oeni* et leur adaptation au vin.

La forte abondance des pseudogènes confirme que le répertoire des gènes codant pour des protéines est soumis à des altérations. De tels traits d'évolution sont rarement rencontrés dans la biosphère des micro-organismes, excepté chez ceux évoluant dans des niches écologiques stressantes, comme l'espèce *Geobacillus* (vivant dans les sources d'eau chaude). Dans les génomes d'*O. oeni*, les processus d'érosion des gènes sont favorables aux mutations et fortement influencés par la propention naturelle de ces génomes à l'hypermutabilité et à une évolution rapide. De manière intéressante, l'accumulation apparente des pseudogènes sur certaines régions chromosomiques (appelée ici ψ_{oasis}) supporte l'idée d'une organisation spatiale à grande échelle de l'érosion des gènes. Plus particulièrement, nous avons pu mettre en évidence l'architecture du pan-génome (soit du core-pseudome et de l'accessory-pseudome) et un ensemble de règles (spécifiques à une souche ou à une espèce) gouvernant la dynamique d'altération des gènes au sein de ces souches bactériennes spécifiques.

Afin de pouvoir passer à l'échelle, l'automatisation complète de la méthode permettrait l'analyse d'un jeu de données plus complet et de généraliser les résultats obtenus.

Bibliographie

Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped blast and psi-blast : a new generation of protein database search programs. *Nucleic acids research*, 25(17) :3389–3402.

Bartowsky, E. and Borneman, A. (2011). Genomic variations of oenococcus oeni strains and the potential to impact on malolactic fermentation and aroma compounds in wine. *Applied microbiology and biotechnology*, 92(3) :441–447.

Bilhère, E., Lucas, P., Claisse, O., and Lonvaud-Funel, A. (2009). Multilocus sequence typing of oenococcus oeni : detection of two subpopulations shaped by intergenic recombination. *Applied and environmental microbiology*, 75(5) :1291–1300.

Boido, E., Lloret, A., Medina, K., Carrau, F., and Dellacassa, E. (2002). Effect of β-glycosidase activity of oenococcus oeni on the glycosylated flavor precursors of tannat wine during malolactic fermentation. *Journal of agricultural and food chemistry*, 50(8) :2344–2349.

Bon, E., Delaherche, A., Bilhere, E., De Daruvar, A., Lonvaud-Funel, A., and Le Marrec, C. (2009). Oenococcus oeni genome plasticity is associated with fitness. *Applied and environmental microbiology*, 75(7) :2079–2090.

Bon, E., Granvalet, C., Remize, F., Dimova, D., Lucas, P., et al. (2008). Insights into genome plasticity of the wine-making bacterium oenococcus oeni strain atcc baa-1163 by decryption of its whole genome. In *9th Symposium on Lactic Acid Bacteria*.

Borneman, A., Bartowsky, E., McCarthy, J., and Chambers, P. (2010). Genotypic diversity in oenococcus oeni by high-density microarray comparative genome hybridization and whole genome sequencing. *Applied microbiology and biotechnology*, 86(2) :681–691.

Chelo, I., Ze-Ze, L., and Tenreiro, R. (2007). Congruence of evolutionary relationships inside the leuconostoc-oenococcus-weissella clade assessed by phylogenetic analysis of the 16s rrna gene, dnaa, gyrb, rpoc and dnak. *International journal of systematic and evolutionary microbiology*, 57(2) :276.

Deroin, P. (2010). Des pseudogenes pas si pseudo. *Biofutur*, (313).

Dicks, L., Dellaglio, F., and Collins, M. (1995). Proposal to reclassify leuconostoc oenos as oenococcus oeni. *International Journal of Systematic Bacteriology*, 45(2) :395.

Dicks, L., Van Vuuren, H., and Dellaglio, F. (1990). Taxonomy of leuconostoc species, particularly leuconostoc oenos, as revealed by numerical analysis of total soluble cell protein patterns, dna base compositions, and dna-dna hybridizations. *International Journal of Systematic and Evolutionary Microbiology*, 40(1) :83.

Durrens, P. and Sherman, D. (2005). A systematic nomenclature of chromosomal elements for hemiascomycete yeasts. *Yeast*, 22(5) :337–342.

Eisen, J. and Hanawalt, P. (1999). A phylogenomic study of dna repair genes, proteins, and processes. *Mutat. Res*, 435(3) :171–213.

Endo, A. and Okada, S. (2006). Oenococcus kitaharae sp. nov., a non-acidophilic and non-malolactic-fermenting oenococcus isolated from a composting distilled shochu residue. *International journal of systematic and evolutionary microbiology*, 56(10) :2345.

GARVIE, E. (1967). Leuconostoc oenos sp. nov. *Microbiology*, 48(3) :431.

Garvie, E. (1986). Genus leuconostoc. *Bergey's manual of systematic bacteriology*, 2 :1071–1075.

Garvie, E. and Farrow, J. (1980). The differentiation of leuconostoc oenos from non-acidophilic species of leuconostoc, and the identification of five strains from the american type culture collection. *American Journal of Enology and Viticulture*, 31(2) :154.

Grimaldi, A., Bartowsky, E., and Jiranek, V. (2005). A survey of glycosidase activities of commercial wine strains of oenococcus oeni. *International journal of food microbiology*, 105(2) :233–244.

Henick-Kling, T., Acree, T., Krieger, S., Laurent, M., and Edinger, W. (1994). Modification of wine flavor by malolactic fermentation. *Wine East*, 4 :8–15.

Junier, I., Hérisson, J., and Képès, F. (2010). Periodic pattern detection in sparse boolean sequences. *Algorithms for Molecular Biology*, 5 :31.

Koonin, E. (2005). Orthologs, paralogs, and evolutionary genomics 1. *Annu. Rev. Genet.*, 39 :309–338.

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., Jones, S., and Marra, M. (2009). Circos : an information aesthetic for comparative genomics. *Genome research*, 19(9) :1639–1645.

Lerat, E. and Ochman, H. (2004). ψ-ϕ : Exploring the outer limits of bacterial pseudogenes. *Genome research*, 14(11) :2273–2278.

Liu, Y., Harrison, P., Kunin, V., and Gerstein, M. (2004). Comprehensive analysis of pseudogenes in prokaryotes : widespread gene decay and failure of putative horizontally transferred genes. *Genome biology*, 5(9) :R64.

Ludwig, W., Schleifer, K., and Whitman, W. (2009). Revised road map to the phylum firmicutes. *Systematic Bacteriology*, pages 1–13.

Makarova, K. and Koonin, E. (2007). Evolutionary genomics of lactic acid bacteria. *Journal of bacteriology*, 189(4) :1199–1208.

Makarova, K., Slesarev, A., Wolf, Y., Sorokin, A., Mirkin, B., Koonin, E., Pavlov, A., Pavlova, N., Karamychev, V., Polouchine, N., et al. (2006). Comparative genomics of the lactic acid bacteria. *Proceedings of the National Academy of Sciences*, 103(42) :15611.

Marcobal, A., Sela, D., Wolf, Y., Makarova, K., and Mills, D. (2008). Role of hypermutability in the evolution of the genus oenococcus. *Journal of bacteriology*, 190(2) :564–570.

Martineau, B. and Henick-Kling, T. (1995). Performance and diacetyl production of commercial strains of malolactic bacteria in wine. *Journal of Applied Microbiology*, 78(5) :526–536.

Martinez-Murcia, A., Harland, N., and Collins, M. (1993). Phylogenetic analysis of some leuconostocs and related organisms as determined from large-subunit rrna gene sequences : assessment of congruence of small-and large-subunit rrna derived trees. *The Journal of applied bacteriology*, 74(5) :532.

Mathelier, A. and Carbone, A. (2010). Chromosomal periodicity and positional networks of genes in escherichia coli. *Molecular systems biology*, 6(1).

Matic, I., Rayssiguier, C., and Radman, M. (1995). Interspecies gene exchange in bacteria : the role of sos and mismatch repair systems in evolution of species. *Cell*, 80(3) :507–515.

Mills, D., Rawsthorne, H., Parker, C., Tamir, D., and Makarova, K. (2005). Genomic analysis of oenococcus oeni psu-1 and its relevance to winemaking. *FEMS microbiology reviews*, 29(3) :465–475.

Morse, R., Collins, M., O'HANLON, K., Wallbanks, S., and Richardson, P. (1996). Analysis of the {beta}'subunit of dna-dependent rna polymerase does not support the hypothesis inferred from 16s rrna analysis that oenococcus oeni (formerly leuconostoc oenos) is a tachytelic (fast-evolving) bacterium. *International Journal of Systematic and Evolutionary Microbiology*, 46(4) :1004.

Prunier, A. and Leclercq, R. (2005). Role of muts and mutl genes in hypermutability and recombination in staphylococcus aureus. *Journal of bacteriology*, 187(10) :3455–3464.

Rouchka, E. and Cha, I. (2009). Current trends in pseudogene detection and characterization. *Current Bioinformatics*, 4(2) :112–119.

Taly, J., Magis, C., Bussotti, G., Chang, J., Di Tommaso, P., Erb, I., Espinosa-Carrasco, J., Kemena, C., and Notredame, C. (2011). Using the t-coffee package to build multiple sequence alignments of protein, rna, dna sequences and 3d structures. *nature protocols*, 6(11) :1669–1682.

Van de Guchte, M., Penaud, S., Grimaldi, C., Barbe, V., Bryson, K., Nicolas, P., Robert, C., Oztas, S., Mangenot, S., Couloux, A., et al. (2006). The complete genome sequence of lactobacillus bulgaricus reveals extensive and ongoing reductive evolution. *Proceedings of the National Academy of Sciences*, 103(24) :9274.

YANG, D. and Woese, C. (1989). Phylogenetic structure of the leuconostocs : an interesting case of a rapidly evolving organism. *Systematic and applied microbiology*, 12(2) :145–149.

Synthèse & Perspectives

L'analyse de la molécule d'ADN a longtemps été au coeur de nombreux travaux autour de l'étude des fonctions cellulaires. Cependant la compréhension de l'implication de cette molécule est dépendante de la compréhension des phénomènes complexes de régulation. Cette appréhension des mécanismes de la régulation cellulaire constitue depuis une quinzaine d'années l'un des enjeux principaux de la Bioinformatique. Afin de modéliser ces systèmes de régulation, il est au préalable nécessaire de développer les connaissances relatives aux acteurs de ces systèmes. C'est dans ce contexte que nous avons développé deux méthodes, autour de deux de ces acteurs, ayant pour objectifs (1) d'identifier les ARNnc, (2) de détecter les pseudogènes.

RNA-unchained. Lors de nos travaux sur le chaînage, nous avons vu qu'il existait deux principaux algorithmes de chaînage 2D sur les séquences, un par balayage et un par programmation dynamique. Ces deux algorithmes ont pu être fusionnés en un algorithme hybride tirant parti de chacun des deux algorithmes dont il est issu.

De manière parallèle, on dénombre deux principaux algorithmes de chaînage 2D sur les arborescences, toujours un par balayage et un par programmation dynamique. Tout comme il est possible d'implémenter un algorithme hybride de comparaison de séquences, il peut être envisagé de combiner les deux algorithmes sur les arborescences pour en extrapoler un algorithme hybride qui, comme son analogue sur les séquences, tire avantage de chacun des deux algorithmes dont il est issu.

Nous avons également présenté un filtre qui permet la comparaison en séquence et en structure d'un ARN avec un ensemble d'ARN. Cette méthode se base sur un ensemble de notions : (1) les graines centrées réduites de paramètres l et d variables, (2) l'indexation de graines, réalisée une unique fois en un temps linéaire, et (3) le chaînage rapide de hits grâce à un algorithme de chaînage 2D par balayage sur les arborescences et de complexité sub-cubique.

La comparaison d'*RNA-unchained* avec d'autres outils à partir d'expériences sur le benchmark BraliBase2.1 a permis la mise en évidence de résultats comparables mais surtout une amélioration significative de la qualité des alignements produits par *RNA-unchained* pour des séquences présentant une similarité comprise entre

$60\% - 80\%$. Les performances d'*RNA-unchained* pourraient être améliorées suite à sa nouvelle version implémentée en C++.

De manière générale, ce travail et les travaux centrés sur le chaînage de hits suggèrent qu'une telle approche mérite d'être étudiée, tant du point de vue du modèle de hits que du point de vue des algorithmes de chaînage. Afin d'améliorer la qualité des alignements produits par *RNA-unchained* certaines caractéristiques du filtre pourraient être étudiées de manière plus approfondie et certaines pistes sont envisageables.

Tout d'abord, la notion de « graines à trous » pourrait être étendue en faisant varier la position des mesappariements sur la graine selon les différentes combinaisons possibles.

D'autre part, nous avons pu observer l'impact des paramètres l et d des graines employées. Il semble alors naturel de se demander si une étape préalable d'analyse du jeu de données permettant d'établir les valeurs idéales pour l et d n'améliorerait pas la qualité des résultats d'*RNA-unchained*.

RNA-unchained intègre d'ores-et-déjà la possibilité de combiner plusieurs graines de paramètres l et d différents. Une analyse de l'impact de ces combinaisons constituerait une suite logique à ces travaux.

De plus, l'index employé actuellement, la table de hachage, est une structure creuse qui prend une place conséquente en mémoire. Une nouvelle structure d'indexation plus performante permettrait d'améliorer le filtre.

Il serait intéressant d'expérimenter *RNA-unchained* sur un jeu de données biologiques avec une problématique concrète afin d'estimer son utilité et sa pertinence dans un cas d'analyse de données réelles.

RNA-unchained pourrait s'intégrer dans la mise en place d'une plateforme d'identification des ARN de la Rfam ou de manière plus généraliste répondre aux problématiques de « clustering » et d'identification des ARNnc.

L'une des problématiques actuelles concerne l'importante masse de données produite par les analyses par RNAseq. Les séquences ARN générées ne sont pas toujours complètes, une approche telle qu'*RNA-unchained* pourrait permettre de discriminer les séquences générées.

PseudOE. Le pseudome peut être vu comme un ensemble de séquences sujettes à l'exaptation de nouvelles fonctionnalités, mais aussi comme un espace de recherche aussi bien du point de vue de l'évolution que du point de vue algorithmique. L'introspection de cet espace non codant des génomes a pour objectif d'améliorer la sensibilité et la spécificité des méthodes d'identification et de caractérisation automatiques des objets géniques et d'inférer leur contribution à la signature adaptative (biotechnologique, épidémiologique,...) caractérisant les groupes phylogénétiques explorés.

Dans ce but, nous avons mis en place la méthode *PseudOE* à partir des génomes d'*Oenococcus* qui présentent de telles caractéristiques. Cette méthode a permis la

détection de nouveaux pseudogènes candidats et la redéfinition des répertoires géniques des génomes analysés.

Ce nouvel ensemble de pseudogènes candidats a permis d'analyser la prévalence des pseudogènes au sein des bactéries lactiques mais également la diversité des classes pseudogéniques rencontrées.

Ces données ont alors permis une étude comparative afin d'établir l'architecture du pan pseudome et d'inférer leur implication dans les propriétés de ces souches. En particulier, on peut noter que la forte propension de certains génomes à présenter de nombreuses séquences pseudogénisées semble être corrélée à leur mode de vie présentant des caractéristiques extrêmes.

On note également que la répartition des pseudogènes le long du chromosome ne suit pas une répartition aléatoire. Cette certaine « rythmicité » de localisation des pseudogènes semble faire écho aux récentes études concernant la répartition des objets géniques selon le repliement de la molécule d'ADN.

La méthode *PseudOE* s'inscrit dans une analyse à plus grande échelle d'un ensemble de génomes séquencés afin de confirmer et d'affiner les premières constatations quant à l'organisation du pseudome mais aussi afin d'établir un modèle évolutionnaire de la plasticité génique des génomes bactériens.

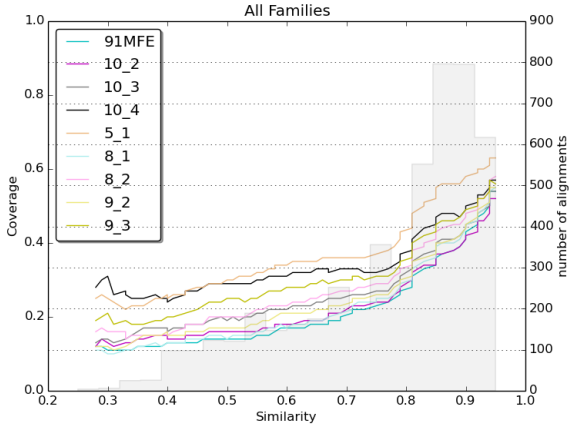
Il est également raisonnable de penser qu'une telle étude contribuera à l'observation de nouveaux marqueurs moléculaires de traits phénotypiques non encore explicités.

Nous avons vu que les pseudogènes peuvent avoir un rôle dans les fonctions cellulaires via leur ARN. Il serait intéressant d'analyser avec *RNA-unchained* les ARN des pseudogènes détectés par la méthode *PseudOE* afin d'identifier s'ils présentent toujours une structure similaire à celle de l'ARN du gène dont dérive le pseudogène ou bien si cette structure est altérée ou impliquée dans la régulation de l'expression génique en tant que leurre ARN. De manière complémentaire, il serait alors intéressant d'analyser la structure des ARN des séquences intergéniques afin d'en inférer la présence de séquences pseudogénisées.

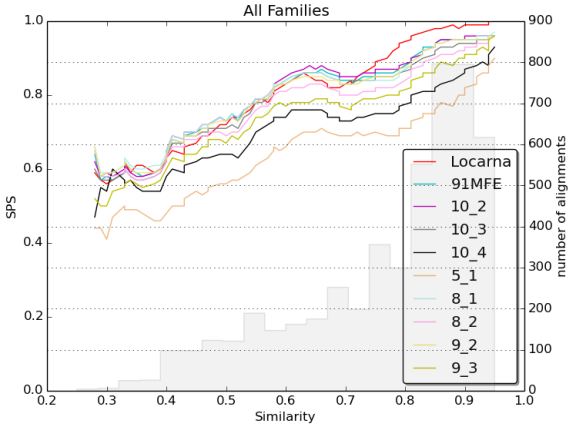
Annexes

Micro-organisme	Souche	Chr. (kb)	Nb total de gènes	Gènes codants	Pseudogènes	ψ prévalence
Enterococcus faecalis	V583	3 218	3 257	3 112	1	0%
Lb lactis	IL1403	2 365	2425	2 321	1	0%
L citrum	KM20	1 796	1785	1 702	1	0.1%
Carnobacterium sp.	17-4	2 635	2 521	2 420	9	0.4%
L. mesenteroïdes	ATCC 8293	2 038	2 073	1 970	19	0.9%
P. pentosaceus	ATCC 25745	1 832	1 847	1 755	20	1.1%
Lb plantarum	WCFS1	3 308	3 135	3 007	42	1.3%
Streptococcus gallolyticus	UCN-34	2 351	2 349	2 223	37	1.6%
Lb reuteri	F275	2 000	2 027	1 900	39	1.9%
Lb brevis	ATCC 367	2 291	2 314	2 218	52	2.2%
Lb gasseri	ATCC 33323	1 894	1 898	1 755	43	2.3%
Lb salivarius	UCC118	1 827	1 864	1 717	49	2.6%
Lb casei	ATCC 334	2 895	2 909	2 751	82	2.8%
Streptococcus suis	P1/7	2 007	2 011	1 824	80	4.0%
Lb cremoris	SK11	2 438	2 610	2 384	144	5.5%
Streptococcus pneumoniae	ATCC 700669	2 221	2 224	1 990	141	6.3%
O. oeni	PSU-1	1780	1 864	1 691	122	6.5%
O. oeni	ATCC BAA-1163	1798	1 880	1 674	155	8.2%
Lb helveticus	DPC 4571	2 080	1 838	1 610	155	8.4%
Lb delbrueckii	ATCC BAA-365	1 857	2 040	1 721	192	9.4%
S. thermophilus	LMD-9	1856	2 003	1 710	206	10.3%

TABLE 7 – *Variation du nombre de pseudogènes dans diverses souches de bactéroes lactiques.*

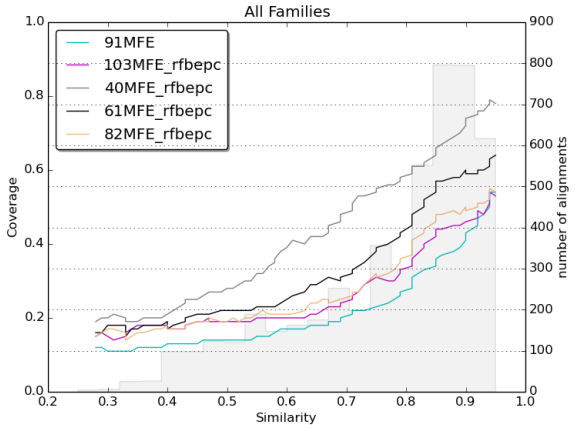


(a) Analyse de la couverture des graines

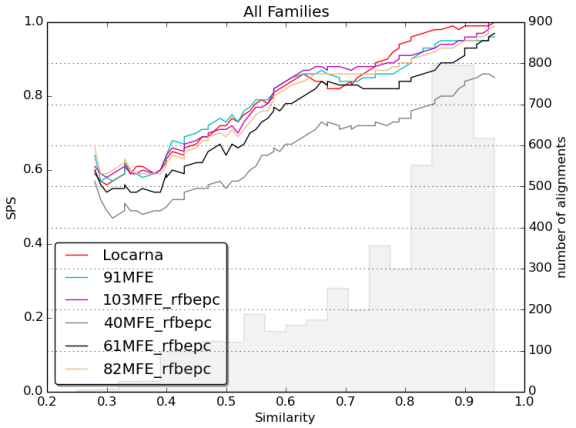


(b) Analyse de la qualité des alignements

Annexe 81 – Impact des différentes valeurs de l et d des graines de RNA-unchained.

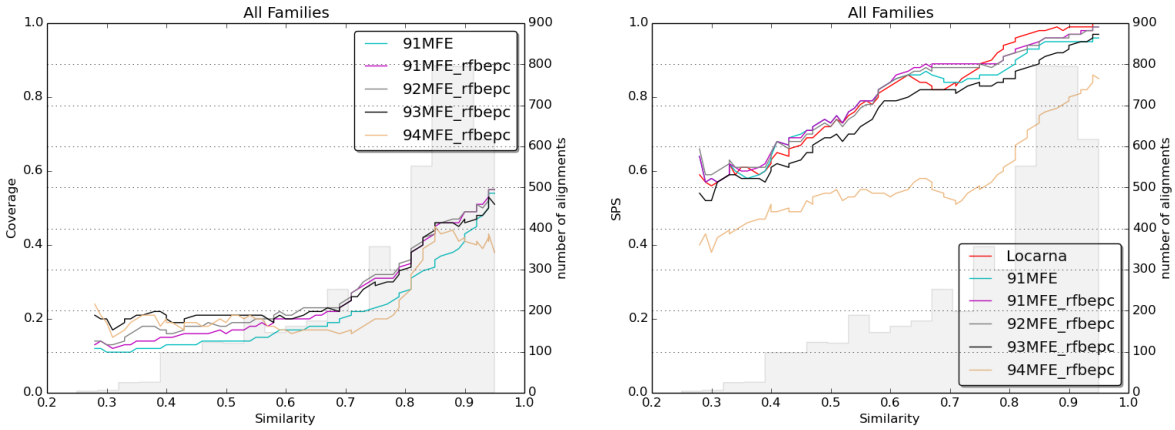


(a) Analyse de la couverture des graines



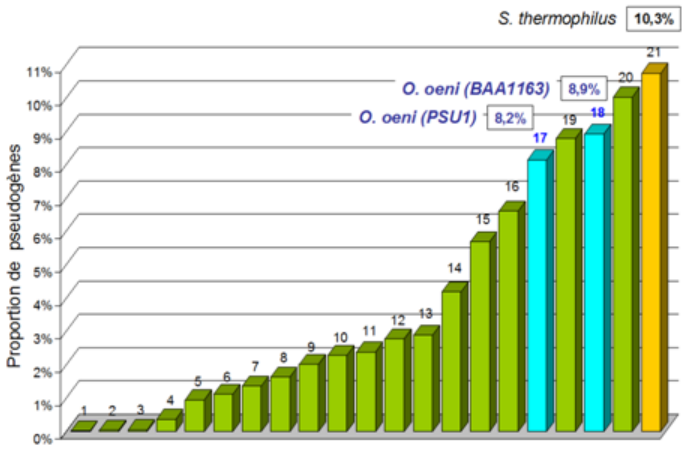
(b) Analyse de la qualitÃ© des alignements

Annexe 82 – Impact des options de RNA-unchained sur diverses tailles de structures pour les graines.



(a) Analyse de la couverture des graines *(b) Analyse de la qualité des alignements*

Annexe 83 – Impact des options de RNA-unchained sur diverses tailles de séquences pour les graines.



	Micro-Organisme	Souche	Accession NCBI	Chromo-some (kb)	GC%	Nb total gènes	Gènes codants protéines	Pseudogenes	
1	*Enterococcus faecalis*	V583	NC_004668	3 218	37,0	3 257	3 112	1	0,0%
2	*Lb lactis*	IL1403	NC_002662	2 365	35,3	2 425	2 321	1	0,0%
3	*L. citrum*	KM20	NC_010471	1 796	39,0	1 785	1 702	1	0,1%
4	*Carnobacterium sp.*	17-4	NC_015391	2 635	35,0	2 521	2 420	9	0,4%
5	*L.mesenteroides*	ATCC 8293	NC_008531	2 038	37,7	2 073	1 970	19	1,0%
6	*P. pentosaceus*	ATCC 25745	NC_008525	1 832	37,4	1 847	1 755	20	1,1%
7	*Lb plantarum*	WCFS1	NC_004567	3 308	44,5	3 135	3 007	42	1,4%
8	*Streptococcus gallolyticus*	UCN-34	NC_013798	2 351	37,0	2 349	2 223	37	1,6%
9	*Lb reuteri*	F275	NC_009513	2 000	38,9	2 027	1 900	39	2,0%
10	*Lb. brevis*	ATCC-367	NC_008497	2 291	46,2	2 314	2 218	52	2,3%
11	*Lb gasseri*	ATCC 33323	NC_008530	1 894	35,3	1 898	1 755	43	2,4%
12	*Lb salivarius*	UCC118	NC_007929	1 827	32,9	1 864	1 717	49	2,8%
13	*Lb casei*	ATCC-334	NC_008526	2 895	46,6	2 909	2 751	82	2,9%
14	*Streptococcus suis*	P1/7	NC_012925	2 007	41,0	2 011	1 824	80	4,2%
15	*Lb cremoris*	SK11	NC_008527	2 438	35,9	2 610	2 384	144	5,7%
16	*Streptococcus pneumoniae*	ATCC 700669	NC_011900	2 221	39,0	2 224	1 990	141	6,6%
17	***O.oeni***	**PSU1**	**NC_008528**	**1 780**	**37,9**	**1 864**	**1 634**	**145**	**8,2%**
19	*Lb helveticus*	DPC 4571	NC_010080	2 080	37,1	1 838	1 610	155	8,8%
18	***O.oeni***	**ATCC BAA-1163**	**Non public**	**1 792**	**37,8**	**1 775**	**1 570**	**154**	**8,9%**
20	*Lb delbrueckii*	ATCC BAA-365	NC_008529	1 857	49,7	2 040	1 721	192	10,0%
21	*S. thermophilus*	LMD-9	NC_00532	1 856	39,1	2 003	1 710	206	10,8%

Annexe 84 – Proportion des pseudogènes au sein de certaines bactéries.

Espèce	Souche	Version	Génome	État	Complet	Dernière MAJ	ID	Génome ID
Oenococcus kitaharae		0	A	x			OEKIT1.0_A	OEOEN1.0
Oenococcus oeni	BAA-1163	0	A	0	01/12/2006	17/01/2007	OEOEN1.0_A	OEOEN1.0
Oenococcus oeni	AWRIB429	0	A	58	03/03/2010	11/03/2010	OEOEN2.0_A0_	OEOEN2.0
Oenococcus oeni	IOEB_89006	0	A	1098	16/07/2008		OEOEN3.0_A0_	OEOEN3.0
Oenococcus oeni	IOEB_89006	1	A	491	28/09/2009		OEOEN3.1_A0_	OEOEN3.1
Oenococcus oeni	IOEB_89006	2	A	349	04/12/2008		OEOEN3.2_A0_	OEOEN3.2
Oenococcus oeni	KM334	0	A	x	projet		OEOEN4.0_A0_	OEOEN4.0
Oenococcus oeni	km383	0	A	x	projet		OEOEN5.0_A0_	OEOEN5.0
Oenococcus oeni	PSU-1	0	A	0	23/10/2010	23/02/2011	OEOEN6.0_A0_	OEOEN6.0
Oenococcus oeni	8413	0	A	101	27/06/2008		OEOEN7.0_A0_	OEOEN7.0
Oenococcus oeni	8413	1	A	48	13/04/2010		OEOEN7.1_A0_	OEOEN7.1
Oenococcus species	JP7.3.6						OESPE1.0_A	OEOEN1.0
Oenococcus species	M7.2.18						OESPE2.0_A	OEOEN2.0
Lactobacillus acidophilus	30SC	0	A	0	11/03/2011	20/05/2011	LAACI1.0_A0_	LAACI1.0
Lactobacillus brevis	ATCC387	0	A	0	21/10/2006	05/03/2010	LABRE1.0_A0_	LABRE1.0
Lactobacillus fermentum	IFO3956	0	A	0	21/04/2008	23/07/2008	LAFER1.0_A0_	LAFER1.0
Lactobacillus plantarum	WCSF1	0	A	11	14/02/2003	29/07/2011	LAPLA1.0_A0_	LAPLA1.0
Lactobacillus reuteri	SD2112	0	A	0	20/06/2011	24/06/2011	LAREU1.0_A0_	LAREU1.0
Lactococcus lactis	subsp. Lactis KF147	0	A	0	23/12/2009	03/05/2010	LALAC1.0_A0_	LALAC1.0

TABLE 8 – *Variation du nombre de pseudogènes dans diverses espèces.*

Espèce	Souche	Version	Génome	État	Complet	Dernière MAJ	ID	Génome ID
Leuconostoc carnosum							LECAR1.0_A	LECAR1.0
Leuconostoc citreum	KM20	0	A	0	17/03/2008	08/04/2008	LECIT1.0_A0_	LECIT1.0
Leuconostoc fallax	KCTC3537	0	A	30	29/12/2010	25/04/2011	LEFAL1.0_A0_	LEFAL1.0
Leuconostoc garlicum							LEGAR1.0_A	LEGAR1.0
Leuconostoc gasicomitatum	LMG18811	0	A	0	30/06/2010		LEGAS1.0_A0_	LEGAS1.0
Leuconostoc gelidum	KCTC3527	0	A	x	projet		LEGEL1.0_A	LEGEL1.0
Leuconostoc holzapelii							LEHOL1.0_A	LEHOL1.0
Leuconostoc inhae	KCTC3774	0	A	x	projet		LEINH1.0_A	LEINH1.0
Leuconostoc kimchii	IMSNU11154	0	A	0	17/05/2010	29/06/2010	LEKIM1.0_A0_	LEKIM1.0
Leuconostoc lactis	KCTC3526	0	A	x	projet		LELAC1.0_A	LELAC1.0
Leuconostoc argentinum	KCTC3773	0	A	98	30/11/2010	22/03/2011	LEARG1.0_A0_	LEARG1.0
Leuconostoc mesenteroides	KFRI-MG	0	A	x	projet		LEMES1.0_A	LEMES1.0
Leuconostoc mesenteroides	subsp. Cremosis ATCC19254	0	A	126	23/04/2009	07/07/2009	LEMES2.0_A0_	LEMES2.0
Leuconostoc mesenteroides	subsp. Dextranicum						LEMES3.0_A	LEMES3.0
Leuconostoc mesenteroides	subsp. Mesenteroides ATCC8293	1	A	0	24/10/2006	13/01/2011	LEMES4.1_A0_	LEMES4.1
Leuconostoc mesenteroides	subps. Mesenteroides Y110	0	A	0	27/02/2004	14/11/2006	LEMES4.0_A0_	LEMES4.0
Leuconostoc palmae							LEPAL1.0_A	LEPAL1.0
Leuconostoc pseudomesenteroides	KCTC3652	0	A	x	projet		LEPSE1.0_A	LEPSE1.0

TABLE 9 – *Variation du nombre de pseudogènes dans diverses espèces (2).*

Espèce	Souche	Version Génome		État	Complet	Dernière MAJ	ID	Génome ID
Pediococcus pentosaceus	ATCC25745	0	A	0	23/10/2006	11/04/2011	PEPEN1.0_A0_	PEPEN1.0
Streptococcus pyogenes	NZ131	0	A	0	16/10/2008	09/12/2008	STPYO1.0_A0_	STPYO1.0
Streptococcus thermophilus	LMG18311	0	A	0	18/11/2004	05/03/2010	STTHE1.0_A0_	STTHE1.0
Weissella koreensis	KACC15510	0	A	0	30/06/2011	07/07/2011	WEKOR1.0_A0_	WEKOR1.0
Weissella paramesenteroides	ATCC33313	0	A	36	22/04/2009	17/07/2009	WEPAR1.0_A0_	WEPAR1.0

TABLE *10 – Variation du nombre de pseudogènes dans diverses espèces (3).*